



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Dynamic Bayesian Models via Monte Carlo - An Introduction with Examples -

G. Johannesson, B. Hanley, J. Nitao

October 12, 2004

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Dynamic Bayesian Models via Monte Carlo

— An Introduction with Examples —

Gardar Johannesson Bill Hanley John Nitao

September 29, 2004

Lawrence Livermore National Laboratory

UCRL-TR-207173

Abstract

This report gives an introduction to a Bayesian probabilistic approach to modeling a dynamic system, with emphasis on stochastic methods for posterior inference. The Bayesian paradigm is a powerful tool to combine observed data along with prior knowledge to gain a current (probabilistic) understanding of unknown model parameters. In particular, it provides a very natural framework for updating the state of knowledge in a dynamic system. For complex systems, such updating needs to be carried out via stochastic sampling of unknown model parameters. An overview is given of the well established Markov chain Monte Carlo (MCMC) approach to achieve this and of the more recent sequential Monte Carlo (SMC) approach, which is better suited for dynamic systems. Examples are provided, including an application to event reconstruction for an atmospheric release.

Contents

1	Short Introduction to Bayesian Modeling	1
1.1	Basic Notation	1
1.2	Bayes' Theory	2
1.3	Examples	2
2	Dynamic Bayesian Models	6
2.1	The Basic Definition	6
2.2	Example: Target Tracking	8
2.3	Example: Atmospheric Dispersion Modeling with Unknown Source Characteristics	9
3	Markov Chain Monte Carlo (MCMC)	11
3.1	The Basics of MCMC	11
3.2	Sequential MCMC via Rejuvenation and Extension	14
3.3	Sequential MCMC via Rejuvenation, Modification, and Extension . .	16
3.4	MCMC Proposal Distributions	18
4	Sequential Monte Carlo (SMC)	20
4.1	Importance Sampling (IS)	20
4.2	The Basics of SMC	21
4.3	SMC via Rejuvenation and Extension	23
4.4	SMC via Rejuvenation, Modification and Extension	27
4.5	Hybrid Methods: MCMC within SMC and MCMC prior to SMC . .	28
4.6	SMC Proposal Distributions	29
5	Applications	31
5.1	Bivariate Gaussian Distribution	31
5.2	Atmospheric Dispersion Modeling with Unknown Source Character- istics	34

1 Short Introduction to Bayesian Modeling

We shall now give a brief introduction to the Bayesian paradigm to modeling and inference, along with examples. A good introduction to Bayesian theory and modeling is “Bayesian Theory” by Bernardo & Smith (1994) and “Bayesian Data Analysis” by Gelman et al. (2004).

1.1 Basic Notation

Let X and Y be two random variables and denote by:

$p(Y)$ = the probability distribution of Y .

$p(X, Y)$ = the joint probability distribution of X and Y .

$p(X | Y)$ = the probability distribution of X conditional on Y .

We shall use the same notation for a continuous random variable, in which case $p(\cdot)$ is referring to a continuous density function, and for a discrete random variable, in which case $p(\cdot)$ is referring to a probability mass function. In addition, we shall not in general distinguish between a (unknown) random variable and a particular value it can take; hence, we use $p(Y)$ to mean both the probability distribution of Y or if Y is known (observed) the probability distribution of Y evaluated at that particular observed value.¹ In the case where we need to distinguish between the two, we write $p(Y = y)$ to mean the probability distribution of the random variable Y evaluated at the value y . Hence, if Y is a discrete random variable, $p(Y = y)$ is the probability of $Y = y$, while if Y is a continuous random variable, $p(Y = y)$ is the probability density function of Y evaluated at y .

There are few basic principles that are used repeatedly in this document:

- (1) If the random variables X and Y are independent, then $p(X, Y) = p(X)p(Y)$.
- (2) Given the joint distribution of X and Y , the *marginal* distribution of Y is given by integrating over X ,

$$p(Y) = \int_{\mathcal{X}} p(dX, Y), \quad \text{where } X \in \mathcal{X}.$$

If X is a discrete random variable with possible values x_1, \dots, x_n , then $p(Y) = \sum_{i=1}^n p(X = x_i, Y)$.

- (3) We have the following relationship between the joint distribution, the conditional distribution, and the marginal distribution:

$$p(X, Y) = p(X | Y)p(Y) = p(Y | X)p(X).$$

¹This is a slight abuse of notation, but has become an accepted practice in statistical literature, particularly in Bayesian text.

1.2 Bayes' Theory

Reverend Thomas Bayes' (1702–1761) theory simply states how one can relate the probability of an event X occurring, conditionally on the fact that another event Y has occurred, to the probability of event Y occurring, conditionally on the fact that event X has occurred. Bayes' theory can be written as

$$p(X | Y) = \frac{p(Y | X)p(X)}{p(Y)} \propto p(Y | X)p(X).$$

In above, one can think of X as representing possible model configurations (parameters) and Y as observed data. Then $p(Y | X)$ describes, in a probabilistic way how the observed data Y is linked to a given model configuration X , and is often referred to as the *likelihood* or the *data model*. The distribution $p(X)$ is referred to as the *prior distribution*, describing in a probabilistic way possible model configurations X prior to seeing the data Y . The end result is the *posterior distribution* of X given the data Y , $p(X | Y)$, which describes possible model configurations given (conditional on) the observed data. Given the posterior distribution, one can plot it (particularly if X is one or two dimensional variable) or compute summary statistics for the distribution. Popular statistics include:

$$\text{Mean: } E(X | Y) = \int_{\mathcal{X}} X p(dX | Y).$$

$$\text{Variance: } \text{var}(X | Y) = \int_{\mathcal{X}} (X - E(X | Y))^2 p(dX | Y).$$

$$\text{Mode: } \arg \max_X p(X | Y).$$

One can contrast Bayes' theory to the more classical approach for inference, where X is thought to be an unknown *deterministic* parameter and often *estimated* using, for example maximum likelihood;

$$\hat{X} = \text{the value of } X \text{ that maximizes } p(Y | X).$$

This gives a single best model configuration that is in compliance with the data (as judged by $p(Y | X)$), while the posterior distribution $p(X | Y)$ assigns a probability density over the different model configurations based on their compliance to the observed data and our prior knowledge of X .

1.3 Examples

We shall now give few examples contrasting the classical and Bayesian approach.

Discrete Probability Space

Assume that our (unknown) state-of-the-system parameter X can only take N different, but known values; say x_1, \dots, x_N . From n independent experiments we observe the data y_1, \dots, y_n . The data is assumed to be related to the unknown system parameter X through a (conditional) probabilistic data model,

$$p(Y_i = y_i | X); \quad i = 1, \dots, n,$$

where Y_i is a random variable representing the outcome of the i -th experiment. Due to the independence of the n experiments, the joint distribution of the data, given the state of the system, is

$$p(\mathbf{Y} = \mathbf{y} | X) = \prod_{i=1}^n p(Y_i = y_i | X)$$

where $\mathbf{Y} = (Y_1, \dots, Y_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$. Given a prior distribution on X , $p(X = x_j); j = 1, \dots, N$, the posterior probability distribution of X is given by

$$p(X = x_j | \mathbf{Y} = \mathbf{y}) = \frac{p(\mathbf{Y} = \mathbf{y} | X = x_j)p(X = x_j)}{\sum_{k=1}^N p(\mathbf{Y} = \mathbf{y} | X = x_k)p(X = x_k)}; \quad j = 1, \dots, N,$$

which is easily computed if one can evaluate $p(Y_i = y_i | X = x_j)$ and $p(X = x_j)$ for $i = 1, \dots, n$ and $j = 1, \dots, N$. Further, if one has very little information *a priori* about which state the system is in, an ideal non-informative prior distribution for X is $p(X = x_j) = 1/N; j = 1, \dots, N$. This prior distribution yields

$$p(X = x_j | \mathbf{Y} = \mathbf{y}) \propto p(\mathbf{Y} = \mathbf{y} | X = x_j); \quad j = 1, \dots, N.$$

The maximum likelihood (ML) estimator of X is given by the state

$$\hat{x} = \arg \max_{x \in \{x_1, \dots, x_N\}} p(\mathbf{Y} = \mathbf{y} | X = x).$$

Hence, the ML estimator is the posterior mode (the value x that gives the highest posterior probability) when X is given a non-informative prior distribution.

Gaussian Distributed Measurements

Assume we have the data y_1, \dots, y_n that are independently distributed according to a Gaussian (normal) distribution with mean μ and variance σ^2 ;

$$y_i \sim \text{Gau}(\mu, \sigma^2), \text{ independently for } i = 1, \dots, n,$$

where “ $\sim \text{Gau}(\mu, \sigma^2)$ ” reads “distributed as Gaussian with mean μ and variance σ^2 ”. Assume further that the variance σ^2 is known, but the mean parameter μ

is unknown and our goal is to conduct inference on μ given the data y_1, \dots, y_n . Classical statistical analysis gives the ML estimator of μ as

$$\hat{\mu} = \bar{y}, \text{ where } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

In the Bayesian framework, assume we assign μ the prior distribution

$$\mu \sim \text{Gau}(\xi, \tau^2), \text{ } \xi \text{ and } \tau^2 \text{ known and given.}$$

The posterior distribution of μ , $p(\mu | \mathbf{y})$, can be shown to be $\text{Gau}(M, V)$ with

$$M = \left(\frac{\bar{y}}{\sigma^2/n} + \frac{\xi}{\tau^2} \right) V \text{ and } V = \left(\frac{1}{\sigma^2/n} + \frac{1}{\tau^2} \right)^{-1}.$$

The posterior mean can be seen to be a weighted average of the empirical average \bar{y} and the prior mean ξ . Note as n gets large (more data sampled), M gets closer to \bar{y} , the ML estimator of μ , and the posterior variance gets closer to σ^2/n (which is the variance of \bar{y}). Similarly, as one lets τ^2 grow larger (yielding effectively a non-informative prior for μ), the same effect is seen.

Numerical (Physical) Model

Assume we have a deterministic numerical (physical) model that predicts n different numerical quantities. Let

$$(F_1(\theta), \dots, F_n(\theta)) = \mathbf{F}(\theta),$$

be the n predicted output quantities from the numerical model when configured according to the parameter θ . An experiment is conducted that gives (observed) measurements y_1, \dots, y_n of the quantities that the numerical model $F(\cdot)$ aims at predicting. The observed data is assumed to be related to the model predictions as follows,

$$y_i = F_i(\theta) + \varepsilon_i; \quad i = 1, \dots, n,$$

where $\varepsilon_1, \dots, \varepsilon_n$ are independent Gaussian distributed measurement errors with zero mean and a known variance σ^2 . The data model above can also be written as

$$y_i \sim \text{Gau}(F_i(\theta), \sigma^2); \quad i = 1, \dots, n,$$

yielding a data model $p(y_i | \theta)$ that is a Gaussian distribution with mean $F_i(\theta)$ and variance σ^2 .

The ML estimator of θ is given by

$$\hat{\theta} = \arg \max_{\theta} p(\mathbf{y} | \theta),$$

where

$$p(\mathbf{y} | \theta) = \prod_{i=1}^n p(y_i | \theta).$$

Depending on how computationally involved the numerical model is, and on the dimension of θ , the above (global) optimization can be difficult to carry out.

Given a prior distribution on θ , $p(\theta)$, the posterior distribution of θ is given by

$$p(\theta | \mathbf{y}) = \frac{p(\mathbf{y} | \theta)p(\theta)}{p(\mathbf{y})},$$

where

$$p(\mathbf{y}) = \int p(\mathbf{y} | \theta)p(d\theta).$$

Again, depending on how computationally involved $F(\cdot)$ is and on the dimensionality of θ , evaluating (numerically) the above integral can be prohibitively expensive. Instead of trying to evaluate the integral, an alternative approach is to generate a collection of realizations from the posterior distribution and use these samples to conduct inference (i.e., compute the mean, variance, etc., of the posterior distribution of θ). Indeed, that is the focus of the remaining portion of this report for the case where the posterior distribution of interest is of a particular dynamic form.

2 Dynamic Bayesian Models

We shall now focus on a particular class of probability models that are dynamic by nature. For this class of models the parameter space of interest is *expanding* with time while more data is gathered. Hence, as each new batch of data arrives our goal is to carry out, or rather update our current probabilistic knowledge of the system, which at that point includes both “old” and “new” parameters. A good introduction to dynamic models is “Bayesian Forecasting and Dynamic Models” by West & Harrison (1997).

2.1 The Basic Definition

Denote by

$\boldsymbol{\theta}_t$ the collection of model parameters associated with time t .

\mathbf{y}_t the collection of (potential) data available at time t .

The relationship between the data and the model parameters is described probabilistically by the time-evolving data-model (the likelihood),

$$p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t}); \quad t = 1, 2, \dots, \quad (1)$$

where we have, without loss of generality, assumed discrete and equal-spaced time points, and where

$$\boldsymbol{\theta}_{1:t} \equiv (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_t).$$

Note that the observed data \mathbf{y}_t do not only depend on the parameters at time t , $\boldsymbol{\theta}_t$, but on the whole time history, $\boldsymbol{\theta}_{1:t}$. The joint distribution of all data observed up to and including time t is given by

$$p(\mathbf{y}_{1:t} | \boldsymbol{\theta}_{1:t}) = \prod_{t'=1}^t p(\mathbf{y}_{t'} | \boldsymbol{\theta}_{1:t'}), \quad (2)$$

where $\mathbf{y}_{1:t} \equiv (\mathbf{y}_1, \dots, \mathbf{y}_t)$.

For Bayesian inference a prior distribution is specified for the model parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_t$. Taking advantage of the dynamic nature of the model, the prior distribution can be written as

$$p(\boldsymbol{\theta}_{1:t}) = p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1) \cdots p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}), \quad (3)$$

where the prior distribution of the model parameters at each time point is specified conditional on the model parameters from the previous time points.

We can summarize our dynamic model as:

$$\begin{aligned} \text{Data-Model:} & \quad p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t}) \\ \text{Parameter-Model:} & \quad p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}), \end{aligned} \quad (4)$$

along with the initial prior distribution $p(\boldsymbol{\theta}_1)$; $t = 1, 2, \dots$. It should be noted that both the data model and the parameter model in (4) can also condition on past data, yielding the more general model:

$$\begin{aligned} \text{Data-Model: } & p(\mathbf{y}_t \mid \boldsymbol{\theta}_{1:t}, \mathbf{y}_{1:t-1}) \\ \text{Parameter-Model: } & p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1}). \end{aligned}$$

However, for the remaining of this document we shall assume (4), but the results presented do apply to the more general model above.

Our goal is to conduct posterior inference on $\boldsymbol{\theta}_{1:t}$ as time evolves and more data is gathered. Bayes' theory gives the posterior distribution at time t as

$$\pi_t(\boldsymbol{\theta}_{1:t}) \equiv p(\boldsymbol{\theta}_{1:t} \mid \mathbf{y}_{1:t}) \propto p(\mathbf{y}_{1:t} \mid \boldsymbol{\theta}_{1:t})p(\boldsymbol{\theta}_{1:t}). \quad (5)$$

Using the product form of the likelihood in (2) and the dynamic nature of the prior in (3), we can write the posterior as (or rather, proportional to) the following product,

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto \left(\prod_{t'=1}^t p(\mathbf{y}_{t'} \mid \boldsymbol{\theta}_{1:t'}) \right) \left(\prod_{t'=1}^t p(\boldsymbol{\theta}_{t'} \mid \boldsymbol{\theta}_{1:t'-1}) \right) = \prod_{t'=1}^t p(\mathbf{y}_{t'} \mid \boldsymbol{\theta}_{1:t'})p(\boldsymbol{\theta}_{t'} \mid \boldsymbol{\theta}_{1:t'-1}), \quad (6)$$

where we for convenience define $\boldsymbol{\theta}_{1:0} = \emptyset$ (an empty set of parameters), so that $p(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_{1:0}) = p(\boldsymbol{\theta}_1)$. The above expression for the posterior hints at an alternative, sequential expression for the posterior distribution at time t , that is based on “updating” the posterior distribution from the previous time point, $t - 1$;

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto (p(\mathbf{y}_t \mid \boldsymbol{\theta}_{1:t})p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1})) \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}). \quad (7)$$

One can also derive this posterior updating expression from a purely statistical argument as follows: Given $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$, our prior knowledge of $\boldsymbol{\theta}_{1:t}$ at time t based on all data up to and including time $t - 1$ is given by the distribution

$$\begin{aligned} \pi_{t-1}(\boldsymbol{\theta}_{1:t}) & \equiv p(\boldsymbol{\theta}_{1:t} \mid \mathbf{y}_{1:t-1}) = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1})p(\boldsymbol{\theta}_{1:t-1} \mid \mathbf{y}_{1:t-1}) \\ & = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1})\pi_{t-1}(\boldsymbol{\theta}_{1:t-1}), \end{aligned}$$

where we used that $\boldsymbol{\theta}_t$ is independent of the data $\mathbf{y}_{1:t-1}$ given the parameter history $\boldsymbol{\theta}_{1:t-1}$ (i.e., $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_{1:t-1}) = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1})$). Using this, we can write the posterior at time t as

$$\pi_t(\boldsymbol{\theta}_{1:t}) = p(\mathbf{y}_t \mid \boldsymbol{\theta}_{1:t})\pi_{t-1}(\boldsymbol{\theta}_{1:t}).$$

Although one can write down the posterior distribution up to a proportionality constant at each given time point t , using it for inference is altogether another problem. Computing the proportionality constant can be prohibitively difficult as it involves a numerical multi-dimensional integral (integrating $p(\mathbf{y}_{1:t} \mid \boldsymbol{\theta}_{1:t})p(\boldsymbol{\theta}_{1:t})$ with

respect to $\boldsymbol{\theta}_{1:t}$). An alternative is to sample (i.e., generate) realizations from the (unscaled) posterior distribution and use them for inference (i.e., computing means, variances, quantiles, etc.). Even if we could compute the missing proportionality constant, sampling based inference is often the only viable option in summarizing the posterior distribution, especially in high dimensional settings. This is what we shall explore in Section 3 and 4, and in particular how one can construct a sampling procedure that samples from $\pi_1(\boldsymbol{\theta}_1)$, $\pi_2(\boldsymbol{\theta}_{1:2})$, \dots , in a sequential effective way, taking advantage of the dynamic nature of the posterior in (7)

2.2 Example: Target Tracking

A classical example of a dynamic model is 2D target tracking. The goal is to track a moving target and report on its location, $\mathbf{x}_t = (x_{1t}, x_{2t})$, and velocity, $\mathbf{v}_t = (v_{1t}, v_{2t})$, at discrete time points $t = 1, 2, \dots$. A simple dynamic model for $\boldsymbol{\theta}_t = (\mathbf{x}_t, \mathbf{v}_t)$ is given by

$$\begin{aligned}\mathbf{x}_t &= \mathbf{x}_{t-1} + 0.5(\mathbf{v}_{t-1} + \mathbf{v}_t) \\ \mathbf{v}_t &= \mathbf{v}_{t-1} + \boldsymbol{\delta}_t,\end{aligned}$$

which linearly interpolates the velocity vector between time $(t - 1)$ and t and assumes an *auto-regressive* model for the velocity vector, where the rate-of-change (the acceleration) $\boldsymbol{\delta}_t$ is assumed Gaussian with mean zero and known variance-covariance matrix \mathbf{W} . The model can also be written in the matrix form

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{v}_t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix} + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \boldsymbol{\delta}_t.$$

For simplicity, assume that the target tracking data consists of (noise corrupted) position observations, $\mathbf{y}_1, \mathbf{y}_2, \dots$, that are related to the actual location of the target via

$$\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\varepsilon}_t,$$

where $\boldsymbol{\varepsilon}_t$ is a zero mean Gaussian measurement error with variance-covariance matrix \mathbf{V} .

Given an initial Gaussian prior distribution for $(\mathbf{x}_1, \mathbf{v}_1)$, a closed-form solution (the *Kalman-filter*) exists for updating the posterior distribution at time $(t - 1)$ to yield the posterior distribution at time t in the form of a Gaussian distribution (see e.g., West & Harrison, 1997, for an overview). This result depends on the linearity of both the measurement model and the dynamic model for $(\mathbf{x}_t, \mathbf{v}_t)$, along with the Gaussian assumption made about the measurement errors and the acceleration of the target.

It is relatively easy to extend the above simple target tracking scenario to a more complicated one, where the observed tracking data are not directly (linearly) related to the location of the target and the maneuvering model for the target

is much more complicated. For such a general model, a closed-form solution for updating the posterior distribution at each time t is seldom available and one needs to resort to sampling-based methods for posterior inference.

2.3 Example: Atmospheric Dispersion Modeling with Unknown Source Characteristics

The goal here is to estimate (probabilistically) the location and release rate history of a contaminant into the atmosphere using a numerical atmospheric contaminant dispersion model and relatively few concentration measurements at given sensor locations. In this simple scenario let

$\mathbf{x}_t \in \mathbb{R}^3$ be the location of a point source in the time interval $(t - 1, t]$.

$s_t \in \mathbb{R}^+$ be the source strength (release rate) in the time interval $(t - 1, t]$.

$\boldsymbol{\theta}_t \equiv (\mathbf{x}_t, s_t)$.

Given the source history $\boldsymbol{\theta}_{1:t} = (\mathbf{x}_{1:t}, \mathbf{s}_{1:t})$ we use a rather simple Gaussian puff model, INPUFF (Petersen & Lavdas, 1986), to predict the resulting concentration of the contaminant. Let

$\hat{C}(\mathbf{x}', t') = \hat{C}(\mathbf{x}', t'; \boldsymbol{\theta}_{1:t'})$ be the model predicted contaminant average concentration in $(t' - 1, t']$ at location \mathbf{x}' due to a source with release history given by $\boldsymbol{\theta}_{1:t'}$.

For the dispersion model in question, the predicted concentration $\hat{C}(\mathbf{x}', t')$ can be broken down into additive contributions from each time interval,

$$\hat{C}(\mathbf{x}', t') = \sum_{t=1}^{t'} \hat{G}_{\mathbf{x}_t, t}(\mathbf{x}', t') s_t, \quad (8)$$

where

$\hat{G}_{\mathbf{x}_t, t}(\mathbf{x}', t')$ gives the predicted average concentration in $(t' - 1, t]$ at \mathbf{x}' due to a source at location \mathbf{x} with a release rate of 1 in $(t - 1, t]$ (and zero outside $(t - 1, t]$).

The observed data is assumed to consist of time-averaged concentration measurements at given sensor (monitor) sites. Assuming a network of M sensors at locations $\mathbf{m}_1, \dots, \mathbf{m}_M$, let

$c_{j,t}$ = the average observed concentration from the j -th sensor in the time interval $(t - 1, t]$.

The observed data is then assumed to be related to the predicted concentration via the simple data model

$$p(c_{j,t} | \hat{C}(\mathbf{m}_j, t)) = \text{Gau}(\hat{C}(\mathbf{m}_j, t), V(\hat{C}(\mathbf{m}_j, t)))|_0^\infty, \quad (9)$$

where $\text{Gau}(\mu, \sigma^2)|_l^u$ denotes a Gaussian (Normal) density with mean μ and variance σ^2 and truncated between l and u ($l < u$), and $V(\cdot)$ is a known variance function.

The model is then fully specified by giving a prior distribution for the source location and the release rate history. We shall assume that the source is not moving, $\mathbf{x}_t = \mathbf{x}$, but little is known about its location. We therefore assign a non-informative prior to the location,

$$p(\mathbf{x}) \propto 1 \text{ if } \mathbf{x} \in \mathcal{X}, 0 \text{ otherwise,}$$

where \mathcal{X} is the spatial domain of interest. The source release is assumed to start at an unknown time $t^* \geq 1$ with a vague information of the initial release rate, but is then assumed to change “smoothly” as time progresses. We formulate this prior information as following:

$$p(t^*) = \begin{cases} 1/t_{\max}^* & \text{if } t^* \in \{1, \dots, t_{\max}^*\}, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

That is, a flat prior on the initial start-time between $t^* = 1$ and $t^* = t_{\max}^*$. For the initial release rate, we assume that

$$p_{t^*}(s_{t^*}) = f_1(s_{t^*}) \quad (11)$$

where $f_1(\cdot)$ is a given prior distribution on positive release and note that $s_1 = \dots = s_{t^*-1} = 0$. And finally for $t > t^*$, we assume that

$$p(s_t | s_{t-1}) = f_2(s_t | s_{t-1}) \quad (12)$$

where $f_2(\cdot | \cdot)$ is a conditional distribution. An example of f_1 and f_2 are:

$$\begin{aligned} f_1(\cdot) &= \text{Gau}(\mu_1, \sigma_1^2)|_0^{c^+} \\ f_2(\cdot | s_{t-1}) &= \text{Gau}(s_{t-1}, \sigma_2^2)|_0^{c^+}, \end{aligned}$$

where the parameters μ_1 , σ_1^2 , and σ_2^2 are assumed known and recall that $\text{Gau}(\cdot, \cdot)|_0^{c^+}$ denotes a truncated Gaussian distribution.

Due to how complicated the model is, particularly the dependence of the dispersion model on the location parameter \mathbf{x} , sampling-based methods need to be used for posterior inference at each time point t . And this is what we shall now study for the dynamic model in general.

3 Markov Chain Monte Carlo (MCMC)

We shall now give a review of the well established Markov chain Monte Carlo (MCMC) approach for generating realizations from the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ in (7); $t = 1, 2, \dots$. A good practical introduction to MCMC is the volume "Markov Chain Monte Carlo in Practice", edited by Gilks et al. (1996), the book "Monte Carlo Strategies in Scientific Computing" by Liu (2001), and the overview paper by Andrieu et al. (2003).

Our basic goal is to generate realizations, $\boldsymbol{\theta}_{1:t}^{(1)}, \dots, \boldsymbol{\theta}_{1:t}^{(N)}$ from the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ in (7) for $t = 1, 2, \dots$. All inference are then conducted using these realizations. That is, for example if $Q(\boldsymbol{\theta}_{1:t})$ is a function of the unknown parameters, then its posterior expected value,

$$E(Q(\boldsymbol{\theta}_{1:t}) | \mathbf{y}_{1:t}) \equiv \int Q(\boldsymbol{\theta}_{1:t}) \pi_t(d\boldsymbol{\theta}_{1:t}),$$

is approximated by

$$\hat{E}(Q(\boldsymbol{\theta}_{1:t}) | \mathbf{y}_{1:t}) \equiv \sum_{i=1}^N (1/N) Q(\boldsymbol{\theta}_{1:t}^{(i)}).$$

Basically we have approximated the posterior distribution at time t , $\pi_t(\boldsymbol{\theta}_{1:t})$, by the empirical distribution function,

$$\hat{\pi}_t^N(\boldsymbol{\theta}_{1:t}) = \sum_{i=1}^N (1/N) \delta(\boldsymbol{\theta}_{1:t}^{(i)} - \boldsymbol{\theta}_{1:t}), \quad (13)$$

where $\delta(\boldsymbol{\theta}_{1:t}^{(i)} - \boldsymbol{\theta}_{1:t}) = 1$ if $\boldsymbol{\theta}_{1:t}^{(i)} = \boldsymbol{\theta}_{1:t}$, otherwise 0.

3.1 The Basics of MCMC

The MCMC approach has a long and successful history for non-dynamic models, but has been shown to be somewhat less appropriate for dynamic models (in its most general form). However, there are cases when MCMC is well suited for dynamic models, one being when the main interest is on a single time point given fixed set of data and as such the model can simply be treated as static.

The MCMC approach generates realization(s) from a Markov chain that has the posterior distribution $\pi_t(\boldsymbol{\theta}_{1:t})$ as its stationary distribution. This is accomplished by generating the realization $\boldsymbol{\theta}_{1:t}^{(i)}$ using the previous realization, $\boldsymbol{\theta}_{1:t}^{(i-1)}$ along with a probabilistic proposal mechanism that outlines how this is done. One of the most popularized MCMC algorithm to generate a chain of size N from $\pi_t(\boldsymbol{\theta}_{1:t})$ is given in Table 1 and is referred to as the *Metropolis-Hastings* (M-H) MCMC sampling algorithm. The proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)})$ in Step A of the M-H algorithm

is specified by the user and can be very general (see Section 3.4). The acceptance ratio in Step B of the M-H algorithm is given by the posterior ratio multiplied by the proposal ratio (or rather divided by the proposal ratio). The inclusion of the proposal ratio is to correct for “bias” in the proposal distribution; note that if the proposal distribution is symmetric (unbiased), that is, $q_t(\tilde{\theta}_{1:t} | \theta_{1:t}^{(i)}) = q_t(\theta_{1:t}^{(i)} | \tilde{\theta}_{1:t})$, then the proposal ratio is just equal to 1 and does not enter the expression for the acceptance ratio.

Table 1: Markov Chain Monte Carlo (MCMC) Algorithm:

The following algorithm describes how to generate realizations from $\pi_t(\theta_{1:t})$ for a given t (i.e., at a given time point).

Step 0 (Initialization): A starting value $\theta_{1:t}^{(1)}$ for the Markov chain is proposed.

For $i = 1, \dots, N - 1$: Use Metropolis-Hasting (M-H) sampler:

Step A (Proposal) Given the i -th step of the Markov chain, $\theta_{1:t}^{(i)}$, the next step is proposed via a *proposal distribution*;

$$\tilde{\theta}_{1:t} \sim q_t(\tilde{\theta}_{1:t} | \theta_{1:t}^{(i)}). \quad (14)$$

Step B (M-H Acceptance Ratio): The acceptance ratio,

$$\rho_t(\tilde{\theta}_{1:t}; \theta_{1:t}^{(i)}) = \frac{\pi_t(\tilde{\theta}_{1:t})q_t(\theta_{1:t}^{(i)} | \tilde{\theta}_{1:t})}{\pi_t(\theta_{1:t}^{(i)})q_t(\tilde{\theta}_{1:t} | \theta_{1:t}^{(i)})} = \frac{p(\mathbf{y}_{1:t} | \tilde{\theta}_{1:t})p(\tilde{\theta}_{1:t})q_t(\theta_{1:t}^{(i)} | \tilde{\theta}_{1:t})}{p(\mathbf{y}_{1:t} | \theta_{1:t}^{(i)})p(\theta_{1:t}^{(i)})q_t(\tilde{\theta}_{1:t} | \theta_{1:t}^{(i)})} \quad (15)$$

is computed, along with the acceptance probability

$$\alpha_t(\theta_{1:t}; \theta_{1:t}^{(i)}) = \min\{\rho_t(\tilde{\theta}_{1:t}; \theta_{1:t}^{(i)}), 1\}.$$

Step C (Selection): Generate $u \sim \text{Uniform}[0, 1]$ and let

$$\theta_{1:t}^{(i+1)} = \begin{cases} \tilde{\theta}_{1:t} & \text{if } u \leq \alpha_t(\tilde{\theta}_{1:t}; \theta_{1:t}^{(i)}), \\ \theta_{1:t}^{(i)} & \text{otherwise.} \end{cases}$$

The efficiency of the M-H algorithm depends on the “quality” of the proposal distribution in Step A. The proposal distribution can be factored in a fashion similar to the prior distribution given in (3). That is (suppressing the chain index i),

$$q_t(\tilde{\theta}_{1:t} | \theta_{1:t}) = q_t(\tilde{\theta}_1 | \theta_{1:t})q_t(\tilde{\theta}_2 | \tilde{\theta}_1, \theta_{1:t}) \cdots q_t(\tilde{\theta}_t | \tilde{\theta}_{1:t-1}, \theta_{1:t}).$$

By restricting the proposal of $\tilde{\theta}_{t'}$, $t' = 1, \dots, t$, to condition only on parameters up to and including time t' (a rather natural restriction given the dynamics of the model), the above proposal distribution can be written as

$$q_t(\tilde{\theta}_{1:t} | \theta_{1:t}) = \prod_{t'=1}^t q_t(\tilde{\theta}_{t'} | \tilde{\theta}_{1:t'-1}, \theta_{1:t'}), \quad (16)$$

where recall that $\theta_{1:0} = \emptyset$.

A typical MCMC proposal algorithm alternates between different type of proposals in a systematic or random fashion with each proposal only modifying a subset of the parameters. For example, a (sub-)proposal distribution that only modifies the t -th parameter (the last parameter) can be written as

$$q_t(\tilde{\theta}_{1:t} | \theta_{1:t}) = q_t(\tilde{\theta}_t | \theta_{1:t}) \delta(\tilde{\theta}_{1:t-1} - \theta_{1:t-1}).$$

Similar sub-proposal distributions can be created for the other components of the parameter vector, and the proposal step A in the MCMC Algorithm in Table 1 would alternate between different sub-proposals.

The acceptance ratio (15) can be written as the product,

$$\rho_t(\tilde{\theta}_{1:t}; \theta_{1:t}) = \prod_{t'=1}^t \left(\frac{p(\mathbf{y}_{t'} | \tilde{\theta}_{1:t'}) p(\tilde{\theta}_{t'} | \tilde{\theta}_{1:t'-1}) q_t(\theta_{t'} | \theta_{1:t'-1}, \tilde{\theta}_{1:t'})}{p(\mathbf{y}_{t'} | \theta_{1:t'}) p(\theta_{t'} | \theta_{1:t'-1}) q_t(\tilde{\theta}_{t'} | \tilde{\theta}_{1:t'-1}, \theta_{1:t'})} \right), \quad (17)$$

using the conditional format of the proposal distribution in (16) and the product format of the posterior in (6). Depending on the proposal distribution, it is not necessarily the case that all the components in the above expression need to be evaluated. For example, if the new proposal $\tilde{\theta}_{1:t}$ is such that only changes are made to $\tilde{\theta}_{t''}$, where $t \geq t'' \geq 1$, then only terms with $t' \geq t''$ in the final product in (17) need to be evaluated, the other terms cancel out.

There are two characteristics that determine the effective sample size (the statistical efficiency) of the MCMC realizations $\theta_{1:t}^{(1)}, \dots, \theta_{1:t}^{(N)}$: the burn-in period and the chain's auto-correlation. The burn-in period represents the number of samples needed at the beginning for the Markov chain to actually reach the state where it is sampling from the target distribution, $\pi_t(\theta_{1:t})$. These initial samples are discarded and not used for inference; hence reducing the effective sample size. The second issue is auto-correlation. Due to the Markovian nature of the algorithm, the realizations $\theta_{1:t}^{(1)}, \dots, \theta_{1:t}^{(N)}$ are not an independent sample from $\pi_t(\theta_{1:t})$; nearby realizations can be highly correlated. The amount of auto-correlation in the sample depends on how well the proposal distribution is able to “mix” the sample and the acceptance rate associated with the proposal distribution. If the proposal distribution alters the chain too little at each step ($\tilde{\theta}_{1:t}$ too close to $\theta_{1:t}$), the resulting MCMC sample tends to show high auto-correlation. Similarly, a proposal distribution that makes large changes at each step typically has a low acceptance ratio and therefore

stays in the same state for a long period of time, which causes high auto-correlation in the final sample. The optimal proposal distribution is somewhere in between, and as a rule of thumb, an acceptance rate around 25% is thought to be good in multi-dimensional problems (Gelman et al., 2004, page 306) (if higher, the proposal distribution is making changes that are too small while if lower, the proposal distribution is making changes that are too big).

The main drawback of the MCMC algorithm for dynamic models is it does not have a natural way of carrying the posterior information available from the sample $\boldsymbol{\theta}_{1:t}^{(1)}, \dots, \boldsymbol{\theta}_{1:t}^{(N)}$ over to time $t+1$, to generate the sample $\boldsymbol{\theta}_{1:t+1}^{(1)}, \dots, \boldsymbol{\theta}_{1:t+1}^{(N)}$. At time $t+1$ one would simply start a new Markov chain, with $\pi_{t+1}(\boldsymbol{\theta}_{1:t+1})$ as its targeting distribution, without taking any direct advantage of the sequential nature of the posterior distribution at time $t+1$, as given by (7). There is one exception to this that applies to a particular MCMC algorithm, an algorithm that rejuvenates and extends the MCMC realizations from the previous time point, which we shall now describe.

3.2 Sequential MCMC via Rejuvenation and Extension

We shall now give a short account of a particular MCMC algorithm that takes advantage of the MCMC realizations from previous time step. We shall see later that this MCMC algorithm mirrors (and in many ways inspires) a very similar Sequential Monte Carlo (SMC) algorithm; see Section 4.3.

Assume at time $t-1$ we have an MCMC sample $\boldsymbol{\theta}_{1:t-1}^{(1)}, \dots, \boldsymbol{\theta}_{1:t-1}^{(N)}$ from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})^2$. Using this sample we derive the following approximation (as in (13)),

$$\pi_{t-1}(\boldsymbol{\theta}_{1:t-1}) \simeq \hat{\pi}_{t-1}^N(\boldsymbol{\theta}_{1:t-1}) \equiv \sum_{i=1}^N (1/N) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}).$$

By plugging this approximation in place of $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ in (7), we derive the following approximation to the posterior at time t ,

$$\begin{aligned} \pi_t(\boldsymbol{\theta}_{1:t}) &\simeq C \times p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t}) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}) \sum_{i=1}^N (1/N) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}) \\ &= C \times \sum_{i=1}^N p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}) (1/N) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}), \end{aligned} \tag{18}$$

where C is an unknown normalizing constant. The approach we take here is to generate samples from the approximation above instead of $\pi_t(\boldsymbol{\theta}_{1:t})$. By taking this approach, we have restricted the to-be-generated realizations from the posterior at time t to be of the form $\boldsymbol{\theta}_{1:t} = (\boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t)$, where $\boldsymbol{\theta}_{1:t-1}^{(I)}$, $I \in \{1, \dots, N\}$, is a

²Assume also that this sample has been corrected for a burn-in period

realization from the posterior at time $t - 1$. Hence, we are simply rejuvenating and extending the past realizations based on the information content of the new data, \mathbf{y}_t . The drawback of this approach is that if the new data is highly informative and not very much in line with what the previous data have indicated, the past posterior sample might not be rich enough (e.g., not large enough) to include a sufficient number of past realizations that are in a good agreement with the new data. Hence, taking this approach usually requires a large number of MCMC realizations (a large N), and even if that is satisfied, it often yields an impoverished sample for conducting inference on $\boldsymbol{\theta}_{t'}$ when $t - t'$ is large.

A well known trick to sample from a *mixture* of distributions, like the one in (18), is to augment the parameter space to include the mixture index; work with $(\boldsymbol{\theta}_{1:t}, I)$ instead of only $\boldsymbol{\theta}_{1:t}$ where $I \in \{1, \dots, N\}$ is the mixture component index. Define the following two distributions associated with the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$:

$$\begin{aligned}\pi_t^a(\boldsymbol{\theta}_{1:t} | I) &\equiv C \times p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)}), \\ \pi_t^a(I) &\equiv 1/N; \quad I = 1, \dots, N.\end{aligned}$$

The joint distribution of the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$ is then

$$\begin{aligned}\pi_t^a(\boldsymbol{\theta}_{1:t}, I) &= \pi_t^a(\boldsymbol{\theta}_{1:t} | I) \pi_t^a(I) \\ &= C \times p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}) \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)}) (1/N),\end{aligned}$$

and in particular, the marginal distribution of $\boldsymbol{\theta}_{1:t}$ with respect to $\pi_t^a(\boldsymbol{\theta}_{1:t}, I)$ is

$$\pi_t^a(\boldsymbol{\theta}_{1:t}) = \sum_{I=1}^N \pi_t^a(\boldsymbol{\theta}_{1:t} | I) \pi_t^a(I) = \text{the mixture in (18)}.$$

This suggests that one could construct a MCMC algorithm to sample from $\pi_t^a(\boldsymbol{\theta}_{1:t}, I)$ and then simply drop the index I , yielding a sample from the above marginal distribution which is equal to the target mixture distribution in (18). The proposal for this augmented approach (i.e., Step A in Table 1) would be:

Step A (Augmented Proposal)

- (1) Sample $\tilde{I} \sim \pi_t^a(\tilde{I}) = \text{a uniform distribution on } \{1, \dots, N\}$.
 - (2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)$.
 - (3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, and the augmented proposal is $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$.
-

What is particularly noticeable about the above augmented proposal is it does not depend on $\boldsymbol{\theta}_{1:t}^{(i)}$, the previous realization from the Markov chain. Proposal distributions that have this feature are often referred to as independent M-H proposals.

Since the proposals are independent they can be made in a parallel fashion, as N independent processes. The augmented proposal can be made slightly more general by replacing step **(1)** by the following:

(1') Sample $\tilde{I} \sim q_t(\tilde{I} | \mathbf{y}_t)$, a discrete proposal distribution on $\{1, \dots, N\}$.

Note that this proposal distribution depends on the new data \mathbf{y}_t , allowing for the possibility of using the new data to see which realizations from time $t - 1$ are more fit to be extended to time t and which are not.

The acceptance ratio (Step B in Table 1) for the augmented proposal is given by:

Step B (Augmented M-H Acceptance Ratio) Let $(\boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)}) = (\boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \boldsymbol{\theta}_t^{(i)}, I^{(i)})$ be the previous sample from the Markov chain, then

$$\rho_t(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I}; \boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)}) = \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) q_t(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \mathbf{y}_t)}{p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}, \boldsymbol{\theta}_t^{(i)}) p(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{1:t-1}^{(I^{(i)})}) q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)}.$$

Due to the augmented independent M-H sampler, the above acceptance ratio does not include any mixed terms, terms that include both components from the next proposed state, $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$, and the current state, $(\boldsymbol{\theta}_{1:t}^{(i)}, I^{(i)})$. Even though this is the case, the acceptance process can not be made in parallel, as N independent processes, as in the proposal step³. This is due to what seems to be a rather random use of the i -th sample to compute the acceptance ratio for the new proposal, and therefore influencing if the new state will be accepted or not; recall that the i -th sample had no impact on how the new proposal was generated! One can therefore ask if it is possible to “adapt” this particular MCMC algorithm such that it can be easily conducted in parallel? The answer to that is Sequential Monte Carlo (SMC), which we shall review in Section 4.

3.3 Sequential MCMC via Rejuvenation, Modification, and Extension

What follows is an outline of how one could modify the above approach to also propose changes in the parameter history (i.e., propose changes to $\boldsymbol{\theta}_{1:t-1}$), not simply rejuvenate and extend the previous realizations to time t . However, this extension results in complications that might in some cases reduce its usefulness.

We replace the augmented proposal step from previous section with the following step:

³Although, one could compute in parallel $p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, $p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$, and $q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)$ for all the N different proposals that can be made in parallel (i.e., at the same time as the proposals are made), and then use to compute the acceptance ratio when needed.

Step A (Augmented Proposal 2)

- (1) Sample $\tilde{I} \sim q_t(\tilde{I} | \mathbf{y}_t)$, a distribution on $\{1, \dots, N\}$.
 - (2a) Sample $\tilde{\boldsymbol{\theta}}_{1:t-1} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t)$.
 - (2b) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t)$.
 - (3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\tilde{\boldsymbol{\theta}}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$.
-

This version of the augmented proposal step both modifies the past (selected) realization $\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}$ and extends it to time t . The proposal distribution in step (2a) can be taken to be of the sequential form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}^{(\tilde{I})}, \mathbf{y}_t).$$

Typically we aim only at changing relatively few parameters associated with $\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}$ in the proposal (those parameters that are believed to have the largest impact on the newly observed data \mathbf{y}_t). As such, many of the sub-proposal distributions above put $\tilde{\boldsymbol{\theta}}_{t'} = \boldsymbol{\theta}_{t'}^{(\tilde{I})}$ with probability 1.

The main difference (and added complexity) of this approach versus the previous approach that did not modify the past, is in computing the acceptance ratio ρ . Instead of computing the acceptance ratio with respect to the mixture approximation in (18), yielding a approximate sample from the posterior, we compute the acceptance ratio with respect to the true posterior, hence yielding a sample from the exact posterior distribution. That is, the mixture approximation is only used to construct the proposal. We use therefore (15), or (17), to compute the acceptance ratio, with $q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)})$ in (15) given by

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)}) = q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t) \sum_{\tilde{I}=1}^N q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \mathbf{y}_t) q_t(\tilde{I} | \mathbf{y}_t).$$

A few comments on evaluating the proposal ratio (15). Since the proposal is derived by modifying a realization from time $t-1$, some of the likelihood and prior calculations involved have already been carried out at time $t-1$. Secondly, evaluating the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$ involves summation from $\tilde{I} = 1$ to N over the realizations from time $t-1$, which can be computationally expensive. However, if only a few of the past parameters are modified, most (if not all except one) of the N terms in the sum are equal to zero, making it manageable to evaluate the mixture sum. (For example, if we only modify the component $\boldsymbol{\theta}_{t-1}^{(\tilde{I})}$ of the selected realization from time $t-1$, then only realizations from time $t-1$ which have identical parameter history from time 1 to $t-2$ yield non-zero probability in computing the summation associated with $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$.)

3.4 MCMC Proposal Distributions

Nothing has been said so far on how the proposal distributions (14) of the MCMC algorithm are specified. In general, the only condition that $q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)})$ in (14) needs to satisfy is the rather natural condition that

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)}) > 0 \text{ if and only if } q_t(\boldsymbol{\theta}_{1:t}^{(i)} | \tilde{\boldsymbol{\theta}}_{1:t}) > 0.$$

We shall now briefly mention few approaches that have been used for constructing proposal distributions.

The Gibbs Sampler.

The Gibbs-sampling approach partitions the parameter vector $\boldsymbol{\theta}_{1:t}$ into blocks of related parameters (e.g., into t blocks with each block given by $\boldsymbol{\theta}_{t'}$; $t' = 1, \dots, t$). A proposal is then made by changing the parameters of a single block at a time using the *full conditional distribution* (see below) of the block's parameters as the proposal distribution. To demonstrate, let each parameter block consist of $\boldsymbol{\theta}_{t'}$; $t' = 1, \dots, t$, and we wish to propose a change to the block indexed by $t' \in \{1, \dots, t\}$. The new proposal, $\tilde{\boldsymbol{\theta}}_{1:t}$, is given by

$$\tilde{\boldsymbol{\theta}}_{1:t \setminus t'} = \boldsymbol{\theta}_{1:t \setminus t'} \text{ and } \tilde{\boldsymbol{\theta}}_{t'} \sim \pi_t(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'}),$$

where $\boldsymbol{\theta}_{1:t \setminus t'} \equiv \{\boldsymbol{\theta}_\tau : \tau = 1, \dots, t, \tau \neq t'\}$ and $\pi(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})$ is the full conditional distribution of $\tilde{\boldsymbol{\theta}}_{t'}$, given by

$$\pi_t(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'}) = p(\tilde{\boldsymbol{\theta}}_{t'} | \mathbf{y}_{1:t}, \boldsymbol{\theta}_{1:t \setminus t'}).$$

The acceptance ratio (15) is then given by

$$\begin{aligned} \rho_t(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}) &= \frac{\pi_t(\boldsymbol{\theta}_{1:t \setminus t'}, \tilde{\boldsymbol{\theta}}_{t'}) \pi_t(\boldsymbol{\theta}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})}{\pi_t(\boldsymbol{\theta}_{1:t \setminus t'}, \boldsymbol{\theta}_{t'}) \pi_t(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})} \\ &= \frac{(\pi_t(\boldsymbol{\theta}_{1:t \setminus t'}) \pi_t(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})) \pi_t(\boldsymbol{\theta}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})}{(\pi_t(\boldsymbol{\theta}_{1:t \setminus t'}) \pi_t(\boldsymbol{\theta}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})) \pi_t(\tilde{\boldsymbol{\theta}}_{t'} | \boldsymbol{\theta}_{1:t \setminus t'})} = 1. \end{aligned}$$

Hence, Gibbs-sampler moves are always accepted. The algorithm updates the different parameter blocks in a systematic order or a parameter block is selected randomly and updated.

For complex models, the full conditional proposal distributions needed are not always available in closed form or readily available for sampling. However, one can aim at constructing a proposal distribution q_t that is an approximation to the full conditional distribution (e.g., a Gaussian approximation). In that case, one would need to compute the acceptance ratio as it is not guaranteed to be equal to 1 (i.e., some of the proposal made by the approximation will most likely be rejected).

Random-Walk MCMC

One of the more common way to create a MCMC proposal distribution is via simple random walk. Let $\boldsymbol{\theta}_{1:t}^{(i)}$ be the current state of the Markov chain. A new proposal is generated as

$$\tilde{\boldsymbol{\theta}}_{1:t} = \boldsymbol{\theta}_{1:t}^{(i)} + \boldsymbol{\delta}_{1:t},$$

where $\boldsymbol{\delta}_{1:t} \sim q_t(\boldsymbol{\delta}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i)})$. Hence, a perturbation is made to the current state of the chain. The new proposal is then accepted or rejected in the usual way.

Langevin Diffusion.

Langevin diffusion can be thought of as a special case of a more general hybrid (or rather 'Hamiltonian') Monte Carlo algorithms (see e.g., Liu, 2001, chapter 9) and yields a more effective random-walk procedure.

Let $\boldsymbol{\theta}_{1:t}^{(i)}$ be the current state of the Markov chain. A new proposal is given by

$$\tilde{\boldsymbol{\theta}}_{1:t} = \boldsymbol{\theta}_{1:t}^{(i)} + \frac{1}{2} \frac{\partial \log \pi_t(\boldsymbol{\theta}_{1:t})}{\partial \boldsymbol{\theta}_{1:t}} \bigg|_{\boldsymbol{\theta}_{1:t}^{(i)}} h + h^{1/2} \mathbf{Z}_t,$$

where $\mathbf{Z}_t \sim \text{Gau}(\mathbf{0}, \mathbf{I})$ and h is a provided step-size parameter. Note the use of the gradient of the log-posterior distribution in determine the proposal — the new proposal has a tendency to be closer to the (local) mode of the posterior distribution. The new proposal is then accepted or rejected in the usual way.

Table 2: Importance Sampling (IS) Algorithm.

(1) Generate a sample of size N from the *proposal distribution* $q(\boldsymbol{\theta})$;

$$\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta}), \quad i = 1, \dots, N.$$

(2) Compute the importance weights,

$$\tilde{w}^{(i)} \propto \frac{\pi(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})}, \quad i = 1, \dots, N,$$

and define $w^{(i)} = \tilde{w}^{(i)} / \sum_{j=1}^N \tilde{w}^{(j)}$.

The distribution $\pi(\cdot)$ is then approximated by

$$\hat{\pi}^N(\boldsymbol{\theta}) \equiv \sum_{i=1}^N w^{(i)} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}),$$

which places the probability mass $w^{(1)}, \dots, w^{(N)}$ on the support points $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$.

4 Sequential Monte Carlo (SMC)

Sequential Monte Carlo (SMC) is inherently designed to sample from dynamic posterior distributions, both in terms of leveraging the dynamic nature of the model and also in terms of reusing previous calculations. As SMC is not Markovian, it is inherently parallel; the different Monte Carlo proposals can be generated and evaluated in parallel. A good Introduction to SMC is "Sequential Monte Carlo Methods in Practice" by Doucet et al. (2001) and "Monte Carlo Strategies in Scientific Computing" by Liu (2001). The paper by Arulampalam et al. (2002) gives a tutorial focusing on Bayesian tracking.

4.1 Importance Sampling (IS)

At the core of the SMC approach is the generation of a weighted sample via importance sampling (IS). Suppose one wants to generate a sample of size N from the distribution $\pi(\boldsymbol{\theta})$ without having direct access to an algorithm to do so, but is able to evaluate $\pi(\boldsymbol{\theta})$ up to a proportionality constant. Importance sampling accomplishes this by using a proposal distribution $q(\boldsymbol{\theta})$, that is close to $\pi(\boldsymbol{\theta})$ and from which it is easy to generate samples. The basic algorithm is given in Table 2 on page 20.

The efficiency of the IS algorithm to generate a representative sample from the target distribution, $\pi(\boldsymbol{\theta})$, is judged by how evenly the importance weights $\{\tilde{w}^{(i)}\}$ are

distributed. One measure on the efficiency is the *effective sample size*, defined as

$$\text{ESS} \equiv \frac{1}{\sum_{i=1}^N (w^{(i)})^2}.$$

If all the weights are equal, then $\text{ESS} = N$, and on the other side, if all the weights are equal to zero except one, then $\text{ESS} = 1$.

For posterior inference, where $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y} | \boldsymbol{\theta})p(\boldsymbol{\theta})$ and \mathbf{y} is the observed data, IS is particularly useful. For example, one could take the proposal distribution as the prior distribution, $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, which would result in

$$\tilde{w}^{(i)} = p(\mathbf{y} | \boldsymbol{\theta}^{(i)}), \quad \text{for } \boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}); i = 1, \dots, N.$$

Hence, the weights would be proportional to the likelihood. Note that this might not yield an effective posterior sample (in terms of ESS) and a better proposal distribution might be needed, that is, a distribution that is closer to $p(\mathbf{y} | \boldsymbol{\theta})p(\boldsymbol{\theta})$.

4.2 The Basics of SMC

Sequential Monte Carlo aims at using IS to generate samples from a sequence of distributions, $\pi_1(\boldsymbol{\theta}_1), \pi_2(\boldsymbol{\theta}_{1:2}), \dots$, without needing to start from “scratch” with each new distribution. This makes SMC particularly efficient for dynamically evolving models. The basic steps of the SMC algorithm are given in Table 3 on page 22.

The SMC algorithm is relatively simple, but as in IS, its effectiveness is determined by how good the proposal distribution is in Step A, Table 3, and how computationally feasible it is to evaluate the resulting importance weights in Step B. By taking advantage of the dynamic nature of the model, the proposal distribution can be partitioned in the same sequential fashion as the prior distribution,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_1)q_t(\tilde{\boldsymbol{\theta}}_2 | \tilde{\boldsymbol{\theta}}_{1:1}) \cdots q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) = \prod_{t'=1}^t q_t(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1}), \quad (20)$$

where recall that $\tilde{\boldsymbol{\theta}}_{1:0} = \emptyset$, an empty set of parameters. For the proposal distribution (20), the IS weight (19) can be written as

$$\tilde{w}_{1:t} \propto \prod_{t'=1}^t \left(\frac{p(\mathbf{y}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'})p(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1})}{q_t(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1})} \right) \propto \tilde{w}_{1:t-1} \left(\frac{p(\mathbf{y}_t | \tilde{\boldsymbol{\theta}}_{1:t})p(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})} \right)$$

using the product format of the posterior in (6).

Note, although not directly indicated, all the conditional distributions in (20) may take advantage of the IS from the previous time point, $\boldsymbol{\theta}_{1:t-1}$, and the new data, \mathbf{y}_t ; along the lines of the sequential MCMC algorithms in Section 3.2 and 3.3. This is really the key to the success of SMC for dynamic problems.

Table 3: Sequential Monte Carlo (SMC) Algorithm:

Initialization: Assume at time $t = t_0 \in \{1, 2, \dots\}$ we have an importance sample

$$\Theta_{1:t_0} = \{\theta_{1:t_0}^{(i)}, w_{1:t_0}^{(i)} : i = 1, \dots, N\}$$

from the posterior distribution $\pi_{t_0}(\theta_{1:t_0})$

For $t = t_0 + 1, t_0 + 2, \dots$:

Step A (Proposal)

For $i = 1, \dots, N$, sample

$$\tilde{\theta}_{1:t}^{(i)} \sim q_t(\tilde{\theta}_{1:t}) = q_t(\tilde{\theta}_t | \tilde{\theta}_{1:t-1}) q_t(\tilde{\theta}_{1:t-1})$$

where $q_t(\tilde{\theta}_{1:t})$ is a user-specified *proposal distribution*. Note how the proposal distribution is partitioned into two parts; first $\tilde{\theta}_{1:t-1}$ is sampled from $q_t(\tilde{\theta}_{1:t-1})$ and then $\tilde{\theta}_t$ is sampled from $q_t(\tilde{\theta}_t | \tilde{\theta}_{1:t-1})$.

The key to a good SMC proposal distribution is to leverage (condition on) $\Theta_{1:t-1}$ and the new data \mathbf{y}_t . That is, take

$$q_t(\tilde{\theta}_t | \tilde{\theta}_{1:t-1}) q_t(\tilde{\theta}_{1:t-1}) = q_t(\tilde{\theta}_t | \tilde{\theta}_{1:t-1}, \mathbf{y}_t) q_t(\tilde{\theta}_{1:t-1} | \Theta_{1:t-1}, \mathbf{y}_t).$$

Step B (Importance Weights)

For $i = 1, \dots, N$, evaluate the unscaled *importance weights*,

$$\tilde{w}_{1:t}^{(i)} \propto \frac{\pi_t(\tilde{\theta}_{1:t}^{(i)})}{q_t(\tilde{\theta}_{1:t}^{(i)})} \propto \frac{p(\mathbf{y}_t | \tilde{\theta}_{1:t}^{(i)}) p(\tilde{\theta}_t^{(i)} | \tilde{\theta}_{1:t-1}^{(i)}) \pi_{t-1}(\tilde{\theta}_{1:t-1}^{(i)})}{q_t(\tilde{\theta}_t^{(i)} | \tilde{\theta}_{1:t-1}^{(i)}) q_t(\tilde{\theta}_{1:t-1}^{(i)})} \quad (19)$$

Let,

$$\theta_{1:t}^{(i)} = \tilde{\theta}_{1:t}^{(i)} \quad \text{and} \quad w_{1:t}^{(i)} = \tilde{w}_{1:t}^{(i)} / \sum_{j=1}^N \tilde{w}_{1:t}^{(j)},$$

then, we have the approximation;

$$\pi_t(\theta_{1:t}) \simeq \hat{\pi}_t^N(\theta_{1:t}) \equiv \sum_{i=1}^N w_{1:t}^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)}).$$

Note. Above, $\theta_{1:t}^{(i)}$ is simply put equal to $\tilde{\theta}_{1:t}^{(i)}$, however, often an additional perturbation step is introduced (e.g., a single MCMC step) yielding $\theta_{1:t}^{(i)}$ different from $\tilde{\theta}_{1:t}^{(i)}$.

A natural way to take advantage of the IS from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ is to build a proposal distribution $q_t(\cdot)$ that conditions on a given realization from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ (similar to Section 3.2). Such proposal distribution can be written as

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t-1}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}), \quad (21)$$

where $\boldsymbol{\theta}_{1:t-1} \sim \pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$.

The proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1})$ can be taken to be of the sequentially form (see also (16)),

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}).$$

Hence, the proposal can be considered to consist of three steps: (1) draw a realization from $\pi_{t-1}(\cdot)$, (2) perturbing the drawn realization via $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1})$, and finally (3) extend the perturbed realization by drawing $\tilde{\boldsymbol{\theta}}_t$ from $q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})$. The immediate drawback of this general conditional proposal approach above is that

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \int q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}) \pi_{t-1}(\boldsymbol{\theta}_{1:t-1}) d\boldsymbol{\theta}_{1:t-1} \quad (22)$$

is needed for the evaluation of the importance weight in (19). This integral is rarely available in closed form and often difficult to evaluate directly. However, one has the approximation,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) \simeq q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \sum_{i=1}^N q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(i)}) w_{1:t-1}^{(i)},$$

using the IS approximation $\hat{\pi}_{t-1}^N(\boldsymbol{\theta}_{1:t-1})$ of $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$. Depending on how the proposal $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1})$ is constructed, most of the terms in the summation above might be equal to zero, and only few a terms would need to be summed up (it is computationally expensive to loop through all N terms of the sum to generate a single proposal — recall there are N proposals to be made). We shall now outline SMC algorithms that take this approach and have been showed to be successful in number of cases; see Doucet et al. (2001).

4.3 SMC via Rejuvenation and Extension

This algorithm is the SMC version of the MCMC rejuvenation and extension algorithm in Section 3.2 — or vice versa. We shall first introduce it from a more classical view which is often attributed to Neil Gordon (Gordon et al., 1993) and referred to as Gordon’s bootstrap filter or simply as a particle filter. We then follow up with a generalization due to Pitt and Shephard (Pitt & Shephard, 1999, 2001) which improves on its efficiency and robustness.

Gordon's Bootstrap Filter

Classical applications of the SMC algorithm often have a data model (a likelihood) where the data at time t only depends on the model parameters at time t ,

$$p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t}) = p(\mathbf{y}_t | \boldsymbol{\theta}_t).$$

An example is the target tracking model in Section 2.2. As such, the newly observed data \mathbf{y}_t is mostly informative about $\boldsymbol{\theta}_t$ and carries less information about $\boldsymbol{\theta}_{t-1}, \dots, \boldsymbol{\theta}_1$. In light of this, a good candidate for the conditional proposal distribution in (21) is

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t-1}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}), \quad (23)$$

where $\boldsymbol{\theta}_{1:t-1} \sim \pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$,

which corresponds to taking $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}) = \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1})$ in (21). That is, $\tilde{\boldsymbol{\theta}}_{1:t} = (\boldsymbol{\theta}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$, and only the new addition, $\tilde{\boldsymbol{\theta}}_t$, is generated and the rest is kept identical to $\boldsymbol{\theta}_{1:t-1}$. Note, there is nothing in the above approach that prevents it from being used for the more general data model $p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t})$. However, if the newly acquired data has information that is not very much in line with past data, this approach could yield a large number of SMC realizations with small weights (i.e., a small effective sample size); this issue was also raised in Section 3.2.

To generate a proposal from (23) one would use the IS from $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$, and replace Step A in Table 3 with:

Step A (Rejuvenation and Extension Proposal)

- (1) Sample \tilde{I} from $\{1, \dots, N\}$ with $p(\tilde{I} = j) = w_{1:t-1}^{(j)}$; $j = 1, \dots, N$.
 - (2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$.
 - (3) Let $\tilde{\boldsymbol{\theta}}_{1:t} \equiv (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$.
-

Since the proposal distribution (23) does not modify the past, the integral in (22) does not need to be evaluated, and the marginal proposal distribution needed for the IS weights in (19) is simply given by

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1}).$$

The resulting IS weights in Step B in Table 3 are then given by

$$\tilde{w}_{1:t} \propto \frac{\pi_t(\tilde{\boldsymbol{\theta}}_{1:t})}{q_t(\tilde{\boldsymbol{\theta}}_{1:t})} \propto \frac{p(\mathbf{y}_t | \tilde{\boldsymbol{\theta}}_{1:t}) p(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \pi_{t-1}(\tilde{\boldsymbol{\theta}}_{1:t-1})} = \frac{p(\mathbf{y}_t | \tilde{\boldsymbol{\theta}}_{1:t}) p(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})}{q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})}, \quad (24)$$

and note how the $\pi_{t-1}(\cdot)$ terms cancel out. Gordon et al. (1993) proposed taking $q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1})$ equal to $p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1})$, yielding $\tilde{w}_{1:t} = p(\mathbf{y}_t | \tilde{\boldsymbol{\theta}}_{1:t})$.

The goal in importance sampling is always to construct a proposal distribution that results in weights of similar size, yielding a large effective sample size. For the case above, when we condition on the past, it translates into selecting a good proposal distribution for $\tilde{\boldsymbol{\theta}}_t$. It can be shown that the full conditional distribution $p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_t)$ is the “optimal” proposal distribution for Gordon’s bootstrap filter, since

$$p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}, \mathbf{y}_t) \propto p(\mathbf{y}_t | \tilde{\boldsymbol{\theta}}_{1:t})p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}), \quad (25)$$

yielding $\tilde{w}_{1:t} \propto 1$ in (24).

Although one might be able to generate $\tilde{\boldsymbol{\theta}}_t$ using the optimal proposal distribution (25), the step before, proposing $\tilde{\boldsymbol{\theta}}_{1:t-1}$, is done without taking into account the new data — it is simply generated from the posterior at time $t-1$ using the IS. This can be particularly inefficient if the new data carries information that has large impact on the past. Pitt & Shephard (1999, 2001) improved upon the basic bootstrap filter, by taking a similar approach as discussed in Section 3.2 and 3.3 on sequential MCMC, which we shall outline now.

Pitt’s and Shephard’s Modification

We could have introduced the bootstrap filter by aiming at generating realizations from the following mixture approximation to $\pi_t(\boldsymbol{\theta}_{1:t})$,

$$\begin{aligned} \hat{\pi}_t(\boldsymbol{\theta}_{1:t}) &\equiv C \times p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t})p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}) \sum_{i=1}^N w_{1:t-1}^{(i)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}) \\ &= C \times \sum_{i=1}^N p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}) w_{1:t-1}^{(i)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}), \end{aligned} \quad (26)$$

which is derived from (7) by replacing $\pi_{t-1}(\boldsymbol{\theta}_{1:t-1})$ with its IS approximation $\hat{\pi}_{t-1}^N(\boldsymbol{\theta}_{1:t-1})$, and where C is an unknown normalizing constant. Then, similar to Section 3.2, we introduce the augmented parameter $(\boldsymbol{\theta}_{1:t}, I)$ with the joint distribution

$$\pi_t^a(\boldsymbol{\theta}_{1:t}, I) = C \times p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}^{(I)}) w_{1:t-1}^{(I)} \delta(\boldsymbol{\theta}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(I)}), \quad (27)$$

and we note that for the marginal distribution of $\boldsymbol{\theta}_{1:t}$ we have that

$$\pi_t^a(\boldsymbol{\theta}_{1:t}) = \sum_{I=1}^N \pi_t^a(\boldsymbol{\theta}_{1:t}, I) = \text{the mixture in (26)}.$$

Hence, as mentioned in Section 3.2, this suggests that we could sample from the joint augmented distribution in (27) and then simply drop the index I to derive a sample from (26).

Pitt and Shephard suggested basing the proposal on the augmented distribution

$$q_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I}) = q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) v_{1:t-1}^{(\tilde{I})} \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}),$$

where the weights $v_{1:t-1}^{(1)}, \dots, v_{1:t-1}^{(N)}$ are allowed to depend on the new data \mathbf{y}_t ; if $v_{1:t-1}^{(i)} = w_{1:t-1}^{(i)}$ their approach yields the bootstrap filter. The marginal proposal distribution for $\tilde{\boldsymbol{\theta}}_{1:t}$ is then

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = \sum_{i=1}^N q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}^{(i)}) v_{1:t-1}^{(i)} \delta(\tilde{\boldsymbol{\theta}}_{1:t-1} - \boldsymbol{\theta}_{1:t-1}^{(i)}), \quad (28)$$

which can be compared to (26). An augmented proposal is then simply generated using the following procedure (very similar to the previous one):

Step A (P & S Rejuvenation and Extension Proposal)

- (1) Sample \tilde{I} from $\{1, \dots, N\}$ with $p(\tilde{I} = j) = v_{1:t-1}^{(j)}$; $j = 1, \dots, N$.
 - (2) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$.
 - (3) Let $\tilde{\boldsymbol{\theta}}_{1:t} = (\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t)$, and the augmented proposal is $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$.
-

The IS weight associated with the proposal $(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})$ is given by

$$\tilde{w}_{1:t} \propto \frac{\pi_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})}{q_t^a(\tilde{\boldsymbol{\theta}}_{1:t}, \tilde{I})} \propto \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) \tilde{w}_{1:t-1}^{(\tilde{I})}}{q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) v_{1:t-1}^{(\tilde{I})}} \quad (29)$$

In light of this, Pitt and Shephard proposed taking

$$v_{1:t-1}^{(i)} \propto w_{1:t-1}^{(i)} p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \hat{\boldsymbol{\theta}}_t) \quad \text{and} \quad q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}) = p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}),$$

where $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t(\boldsymbol{\theta}_{1:t-1}^{(i)})$ is some likely value of $\tilde{\boldsymbol{\theta}}_t$ conditional on $\boldsymbol{\theta}_{1:t-1}^{(i)}$ (i.e., the mode, the mean, or other likely value associated with $p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)})$). Alternatively, one could use

$$v_{1:t-1}^{(i)} \propto w_{1:t-1}^{(i)} p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \hat{\boldsymbol{\theta}}_t) p(\hat{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}) \quad \text{and} \quad q_t(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}) = p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_t),$$

and recall that $p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_t) \propto p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t-1}^{(i)}, \tilde{\boldsymbol{\theta}}_t) p(\tilde{\boldsymbol{\theta}}_t | \boldsymbol{\theta}_{1:t-1}^{(i)})$.

Note that the final weights are not all equal, but the newly acquired data impacted the weights $v_{1:t-1}^{(1)}, \dots, v_{1:t-1}^{(N)}$, and therefore which realizations from time $t-1$ were carried on to time t . This is particularly important when the distribution of \mathbf{y}_t is given by $p(\mathbf{y}_t | \boldsymbol{\theta}_{1:t})$, but not by $p(\mathbf{y}_t | \boldsymbol{\theta}_t)$.

4.4 SMC via Rejuvenation, Modification and Extension

We shall now extend the SMC approach introduced in the previous section to the case where we not only extend previous IS realizations, but also modify them to some extent. (See Section 3.3 for a similar topic.)

One approach to extend previous SMC approaches, that only rejuvenate and extend previous IS realizations, is to consider a proposal distribution of the following form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t}) = q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) \sum_{i=1}^N q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(i)}) q_t(i) \quad (30)$$

where $\{\boldsymbol{\theta}_{1:t-1}^{(i)}\}$ are the IS from time $t-1$. Note that through $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(i)})$ a perturbation can be made to i -th IS realization from time $t-1$. A proposal from this distribution is generated by the following steps:

Step A (Rejuvenation, Modification, and Extension Proposal)

- (1) Sample \tilde{I} from $\{1, \dots, N\}$ with $p(\tilde{I} = i) = q_t(i)$.
 - (2a) Sample $\tilde{\boldsymbol{\theta}}_{1:t-1} \sim q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$.
 - (2b) Sample $\tilde{\boldsymbol{\theta}}_t \sim q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})$.
 - (3) Put $\tilde{\boldsymbol{\theta}}_{1:t} = (\tilde{\boldsymbol{\theta}}_{1:t-1}, \tilde{\boldsymbol{\theta}}_t)$.
-

As mentioned in Section 3.3, the conditional proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$ can be taken to be of the sequential form,

$$q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}) = \prod_{t'=1}^{t-1} q_t(\tilde{\boldsymbol{\theta}}_{t'} | \tilde{\boldsymbol{\theta}}_{1:t'-1}, \boldsymbol{\theta}_{1:t'}^{(\tilde{I})}),$$

and note that each of sub-proposal distributions may depend on the newly acquired data, \mathbf{y}_t . If the new data is only informative for a small subset of the parameters, many of the sub-proposal distributions can simply keep the past value of the parameter intact (i.e., put $\tilde{\boldsymbol{\theta}}_{t'} = \tilde{\boldsymbol{\theta}}_{t'}^{(\tilde{I})}$ with probability 1 for some $t' \in \{1, \dots, t-1\}$).

The main difference between this approach and the previous approach, in which we did not perturb the past IS realizations, is that IS weight calculations can not be based on the IS approximation in (26). The weight calculations need to be based on the original expression for the posterior distribution; see (6). That makes this approach not as computationally efficient as the previous method, however, it is more flexible. The amount of extra computational effort needed depends on how

extensively the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t-1} | \boldsymbol{\theta}_{1:t-1}^{(\tilde{I})})$ modifies the IS realization $\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}$, generated at the previous time point. Recall, from (6), that the posterior distribution can be written as

$$\pi_t(\boldsymbol{\theta}_{1:t}) \propto \prod_{t'=1}^t p(\mathbf{y}_{t'} | \boldsymbol{\theta}_{1:t'}) p(\boldsymbol{\theta}_{t'} | \boldsymbol{\theta}_{1:t'-1}). \quad (31)$$

Then depending on how extensively $\boldsymbol{\theta}_{1:t-1}^{(\tilde{I})}$ is modified by the proposal process, most of the computations involved in computing the above posterior have already been carried out at the previous time point. A similar argument applies to the evaluation of the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_{1:t})$, given by (30) and needed in (19) to compute the IS weight; see also Section 3.3 on generalizing the rejuvenating and extending MCMC algorithm. We shall come back to the issue of modifying past IS realizations later in Section 4.5, where we combine SMC with MCMC to perturb past realizations.

Note. For most applications, it is reasonable to assume that the newly acquired data at time t , \mathbf{y}_t , has information content mostly relevant to parameters close to t in time. That is, \mathbf{y}_t has no (or very small) information value for $\boldsymbol{\theta}$ -parameters sufficiently far into the past; for $\boldsymbol{\theta}_{t'}$ where t' is considerably smaller than t . As such, in practice one does not carry around the whole time history of the $\boldsymbol{\theta}$ parameter, but rather a time window of a fixed size (i.e., $\boldsymbol{\theta}_{1:t}$ is replaced with $\boldsymbol{\theta}_{(t-k):t}$ for some k).

4.5 Hybrid Methods: MCMC within SMC and MCMC prior to SMC

There can be some benefits of mixing SMC and MCMC to generate realizations from the posterior. There are really two areas where MCMC could benefit SMC.

MCMC Within SMC

Some recent attempts have been made using a one or more MCMC steps within each SMC step to perturb the current IS (MacEachern et al., 1999; Gilks & Berzuini, 2001; Godsill & Clapp, 2001). For example, in the case where one adopts SMC based on rejuvenation and extension (i.e., a SMC that does not modify the past), one can at the end of each SMC time step apply one or more MCMC steps to each IS realization. The MCMC step can be very general, and in particular one could propose to modify the past, resulting in a SMC-MCMC hybrid algorithm that rejuvenates, extends, and modifies past IS realizations. The most basic algorithm is as follows:

-
- Step 0:** Assume at time $t - 1$ we have the IS $\Theta_{t-1} = \{\boldsymbol{\theta}_{1:t-1}^{(i)}, w_{1:t-1}^{(i)} : i = 1, \dots, N\}$.
- Step 1:** Carry out a SMC step from time $t - 1$ to t (e.g., using Pitt's and Shephard's rejuvenation and extension algorithm from Section 4.3), yielding a new IS $\Theta_t = \{\boldsymbol{\theta}_{1:t}^{(i)}, w_{1:t}^{(i)} : i = 1, \dots, N\}$.
- Step 2:** Use B MCMC steps to perturb the IS:
- For $i = 1, \dots, N$:
- Step 2.1:** Draw $I_i \in \{1, \dots, N\}$ with $p(I_i = k) = w_{1:t}^{(k)}$; $k = 1, \dots, N$, and put $\boldsymbol{\theta}_{1:t}^{(i,0)} = \boldsymbol{\theta}_{1:t}^{(I_i)}$.
- For $j = 1, \dots, B$:
- Step 3.1:** Make a MCMC proposal $\tilde{\boldsymbol{\theta}}_{1:t} \sim q(\tilde{\boldsymbol{\theta}}_{1:t} | \boldsymbol{\theta}_{1:t}^{(i,j-1)})$.
- Step 3.2:** Compute the MCMC acceptance probability $\alpha(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}^{(i,j-1)})$ and put $\boldsymbol{\theta}_{1:t}^{(i,j)} = \tilde{\boldsymbol{\theta}}_{1:t}$ with probability $\alpha(\tilde{\boldsymbol{\theta}}_{1:t}; \boldsymbol{\theta}_{1:t}^{(i,j-1)})$, else $\boldsymbol{\theta}_{1:t}^{(i,j)} = \boldsymbol{\theta}_{1:t}^{(i,j-1)}$.
- Step 3** The new IS is given by $\boldsymbol{\theta}_{1:t}^{(i)} = \boldsymbol{\theta}_{1:t}^{(i,M)}$ with $w_{1:t}^{(i)} = 1/N$ — that is, the sample is equally weighted.
-

There is a variation to the algorithm above where the random draw in **Step 2.1** is simply replaced with $\boldsymbol{\theta}_{1:t}^{(i,0)} = \boldsymbol{\theta}_{1:t}^{(i)}$.

MCMC Prior to SMC

The SMC algorithm in Table 3 needs to be initialized with an IS at time t_0 ; the first time point of data processing. An ideal way to generate this initial sample is via MCMC using data from time $1, \dots, t_0$. The resulting, equally weighted MCMC sample can then be passed on to SMC for processing data from time $t_0 + 1, t_0 + 2, \dots$

4.6 SMC Proposal Distributions

For a SMC algorithm that just rejuvenates and extends past realizations (the bootstrap filter and Pitt's and Shephard's modification), we have already mentioned two natural candidates for the proposal distribution $q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})$:

$$\begin{aligned} q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) &= p(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}), & \text{(the prior)} \\ q_t(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}) &= p(\tilde{\boldsymbol{\theta}}_t | \tilde{\boldsymbol{\theta}}_{1:t-1}, \mathbf{y}_t). & \text{(the full conditional)} \end{aligned}$$

In the case where the full conditional distribution is not available, one can aim at designing a proposal distribution that is an approximation to the full conditional (this mirrors the Gibbs proposal algorithm in MCMC). Popular approximations are

multi-variate Gaussian or t distributions. If the prior distribution $p(\boldsymbol{\theta}_t | \tilde{\boldsymbol{\theta}}_{1:t-1})$ is informative (relatively narrow and well focused) it is often just sufficient to take the proposal distribution equal to the prior distribution, as suggested by Gordon.

In the case when the past SMC realizations are perturbed by carrying out one or more MCMC steps for each realization, as outlined in previous section, all the proposal methods suggested in Section 3.4 apply.

5 Applications

We shall demonstrate the use of MCMC and SMC for two applications. The first one is a linear Gaussian (Normal) model, a combination of the Gaussian example given in Section 1.3, page 3, and the Gaussian target-tracking setup given in Section 2.2. In this case there is an analytic, closed form expression for the posterior distributions of interest, which can be compared to the sample-derived (MCMC/SMC) posterior inference approach. The second application is the atmospheric event reconstruction problem described in Section 2.3. In this case there is no closed-form analytic expression available for posterior inference.

5.1 Bivariate Gaussian Distribution

Our setup is as follows: Assume at “time” 1 we have the unknown system parameter x_1 (e.g., a location of an object) and an observation y_1 that is assumed to be related to x_1 according to the additive measurement-error model

$$y_1 = x_1 + \varepsilon_1, \quad \text{where } \varepsilon_1 \sim \text{Gau}(0, \sigma^2). \quad (32)$$

The measurement-error model can also be written as $y_1 \sim \text{Gau}(x_1, \sigma^2)$. *A priori*, we assume that

$$x_1 = \mu_1 + \delta_1, \quad \text{where } \delta_1 \sim \text{Gau}(0, \tau^2). \quad (33)$$

That is, $x_1 \sim \text{Gau}(\mu_1, \tau^2)$ where both μ_1 and τ^2 are known. Given this setup, Gaussian theory (e.g., West & Harrison, 1997, chapter 17.2) yields:

$$\begin{aligned} (y_1 | x_1) &\sim \text{Gau}(x_1, \tau^2 + \sigma^2) \\ \begin{bmatrix} y_1 \\ x_1 \end{bmatrix} &\sim \text{Gau} \left(\begin{bmatrix} \mu_1 \\ \mu_1 \end{bmatrix}, \begin{bmatrix} \tau^2 + \sigma^2 & \tau^2 \\ \tau^2 & \tau^2 \end{bmatrix} \right) \\ (x_1 | y_1) &\sim \text{Gau}(\mu_1 + \rho^2(y_1 - \mu_1), \tau^2(1 - \rho^2)) \end{aligned}$$

where $\rho^2 = \tau^2 / (\tau^2 + \sigma^2)$. Hence, the posterior distribution of x_1 given y_1 is

$$p(x_1 | y_1), \text{ and is } \text{Gau}(\mu_1 + \rho^2(y_1 - \mu_1), \tau^2(1 - \rho^2)). \quad (34)$$

For the setup above, we generated synthetic data. We assumed that $x_1 = 0$ and generated y_1 according to the measurement-error model in (32) with $\sigma^2 = 1$, yielding

$$y_1 = -0.626, \text{ drawn from } \text{Gau}(x_1 = 0, \sigma^2 = 1).$$

The parameters associated with the prior for x_1 in (33) were taken to be

$$\mu_1 = 0, \text{ and } \tau^2 = 10^2,$$

yielding a rather vague prior information. This yields a Gaussian posterior distribution for x_1 with mean equal to -0.620 and standard deviation equal to 0.990 ;

see (34). We shall now apply MCMC to sample from the posterior distribution and compare to the true distribution.

We applied MCMC using a Gaussian random-walk proposal distribution,

$$\tilde{x}_1 \sim q_1(\tilde{x}_1 | x_1^{(i)}) = \varphi(\tilde{x}_1; x_1^{(i)}, \xi^2),$$

where $\varphi(\tilde{x}_1; x_1^{(i)}, \xi^2)$ denotes the Gaussian density with mean $x_1^{(i)}$ and variance ξ^2 evaluated at \tilde{x}_1 . Since the proposal distribution is symmetric ($q_1(\tilde{x}_1 | x_1^{(i)}) = q_1(x_1^{(i)} | \tilde{x}_1)$), the acceptance ratio is simply given by

$$\rho(\tilde{x}_1; x_1^{(i)}) = \frac{p(y_1 | \tilde{x}_1)}{p(y_1 | x_1^{(i)})} = \frac{\varphi(y_1; \tilde{x}_1, \sigma^2)}{\varphi(y_1; x_1^{(i)}, \sigma^2)}.$$

We generated three MCMC samples, each of size 2,000, using a different value for ξ in the proposal distribution for each sample; $\xi = 0.35, 2.5, 12$. With $\xi = 0.35$ the acceptance rate was around 90% (too high due to too small step-size), with $\xi = 2.5$ the acceptance rate was around 0.43% (which is close to optimal), while for $\xi = 12$ the acceptance rate was around 10% (too low due to too large step-size). Figure 1 summarizes the results from the three chains. We see how the MCMC sample corresponding to $\xi = 2.5$ mixes better than the other two samples, resulting in smaller auto-correlation (i.e., larger effective sample size). A histogram of the sample realizations is seen to match well the true posterior density.

We shall now extend the above example to “time” 2: At time 2 we have the unknown system variable x_2 (e.g., the object moved to a new location) and a new observation y_2 that is assumed to be related to x_2 according to the same additive measurement-error model as before;

$$y_2 = x_2 + \varepsilon_2, \quad \text{where } \varepsilon_2 \sim \text{Gau}(0, \sigma^2).$$

What we know *a priori* is that x_2 is not too far (different) from x_1 . We therefore assume the following conditional prior distribution for x_2 ,

$$x_2 = x_1 + \eta_2, \quad \text{where } \eta_2 \sim \text{Gau}(0, 1).$$

That is, $x_2 \sim \text{Gau}(x_1, 1)$ *a priori*. To generate synthetic data at time 2, we let $x_2 = 1$ and generate the observation y_2 according to the measurement-error model, yielding $y_2 = 1.184$.

Given the new data y_2 we want to derive a sample from the posterior distribution of (x_1, x_2) given (y_1, y_2) ; that is, from $p(x_1, x_2 | y_1, y_2)$. It should be noted, since all the distributions involved are Gaussian, the posterior distribution is available in closed form as multivariate Gaussian (see e.g., West & Harrison, 1997, chapter 17.2). Hence, we can compare our sample to the true posterior, as before.

There are two approaches we can take to generate realizations from $p(x_1, x_2 | y_1, y_2)$: (1) start a new MCMC to generate a sample from $p(x_1, x_2 | y_1, y_2)$ or (2) use the previous MCMC sample as a starting point for a SMC. The first option would be similar

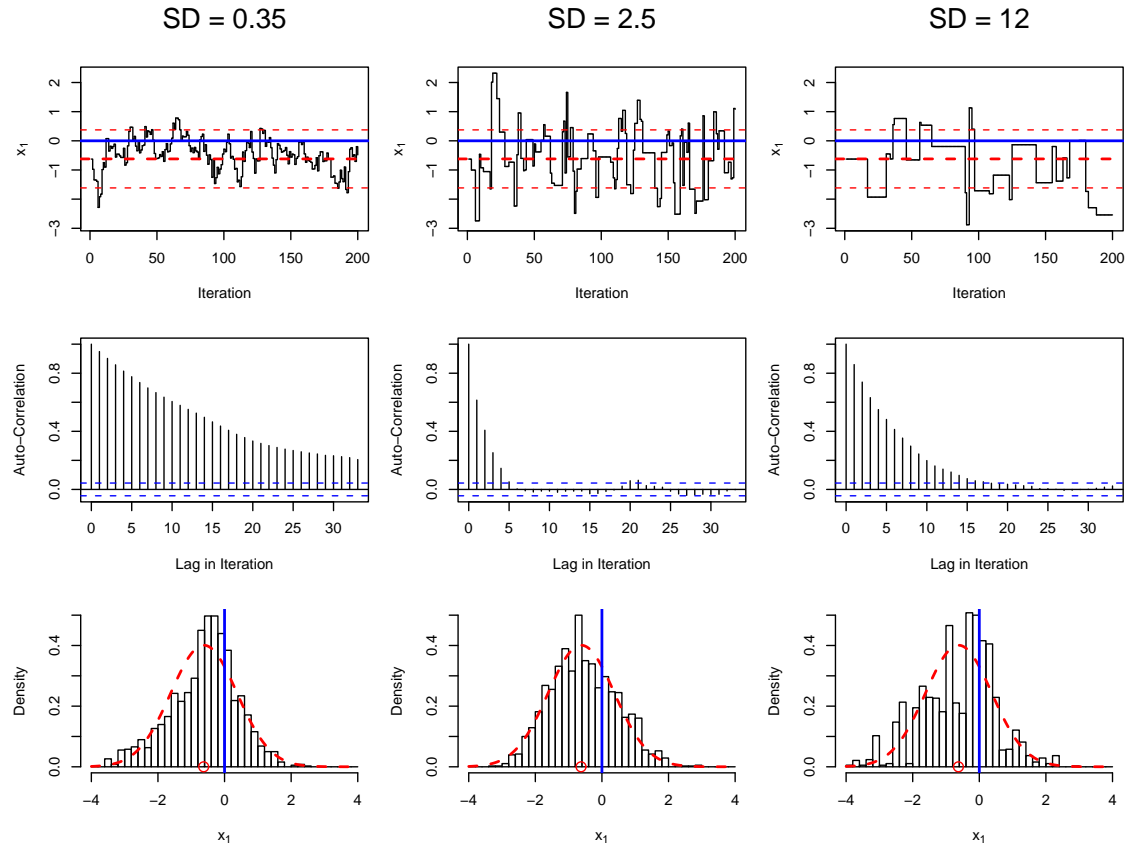


Figure 1: MCMC summary plots for x_1 for three different MCMC samples; left, a proposal distribution with small step-size ($\xi = 0.35$), middle, a proposal distribution with a good step-size ($\xi = 2.5$), and right, a proposal distribution with too big step-size ($\xi = 12$). The first row of plots shows the the first 200 realizations for each chain along with the true value of x_1 superimposed (blue, solid line) and the mean and plus/minus one standard deviation of the true posterior distribution (red,dashed). The middle row of plots shows the auto-correlation in each chain. The bottom row of plots show a histogram of the realizations along with the true value of x_1 (blue, solid) and the true posterior density (red, dashed). The red circles show the data point y_1 .

to the previous MCMC approach for x_1 alone, except we would use, for example a 2D Gaussian random-walk proposal as we have to sample both x_1 and x_2 . We shall therefore demonstrate the use of the second option, SMC, using Gordon's bootstrap filter, as outlined in Table 4 for this implementation.

Table 4: SMC Algorithm for Bivariate Gaussian Example.

Initial Sample: Start with the initial sample $\{x_1^{(i)} : i = 1, \dots, N\}$ generated by MCMC. (Note, it is an equally weighted sample; $w_1^{(i)} = 1/N$.)

For $i = 1, \dots, N$:

- (1) Sample $x_2^{(i)} \sim q_2(x_2 | x_1^{(i)})$, where $q_2(\cdot | x_1^{(i)})$ is $\text{Gau}(x_1^{(i)}, 1)$, the conditional prior distribution.
- (2) Compute the importance weights $\tilde{w}_{1:2}^{(i)} = \varphi(y_2; x_2^{(i)}, \sigma^2)$

The final sample is then given by $\{(x_1^{(i)}, x_2^{(i)}), w_{1:2}^{(i)} : i = 1, \dots, N\}$, where $w_{1:2}^{(i)} = \tilde{w}_{1:2}^{(i)} / \sum_j \tilde{w}_{1:2}^{(j)}$.

To get a final, equally weighted sample at time 2, the final weighted SMC realizations were resampled; that is, 2,000 sample points were drawn from the final collection (with replacement), where the probability of drawing each point is proportional to its final weight. Hence, the resampled collection has multiple copies of realizations with high weights, but realization with low weights have small chance of being picked. (Of the 2,000 original realizations, 1,124 were selected by the resampling process and of those, 500 appeared once in the sample, 372 appeared twice, and 372 three times.) Figure 2 summarizes the results. It shows the marginal histograms of the samples for x_1 and x_2 , along with the true posterior distribution, and the joint distribution of x_1 and x_2 along with true posterior contour lines.

5.2 Atmospheric Dispersion Modeling with Unknown Source Characteristics

We shall now apply MCMC and SMC to estimate an unknown release into the atmosphere using a computer dispersion simulation model as described in Section 2.3.

The Setup: Synthetic Truth and Data

To test the feasibility of using MCMC and SMC to conduct inference on the characteristics of an unknown release into the atmosphere, we generated a synthetic sensor

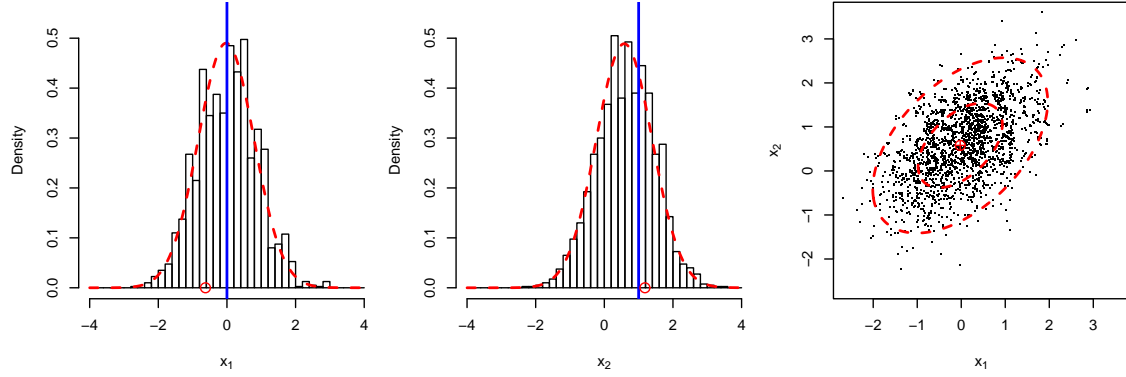


Figure 2: The left and the middle panels show the marginal distribution of x_1 and x_2 , with the true marginal distribution shown as (red) dotted line and the true value of x_1 and x_2 given by a (blue) solid vertical line (a red circles show the observed data). The right panel shows the joint distribution of x_1 and x_2 as represented by the SMC realizations (via resampling). The true mean of the joint posterior distribution is shown along with the 50% and the 95% contour lines.

data from a given source. Our setup is shown in Figure 3 (left). It shows a single stationary source on the left side of the domain, with a constant wind blowing from the West and five sensors located downwind from the source. Our time domain is one hour and is splitted into six 10min intervals. In the first 10min interval the source is not emitting at all, it then emits at a (relative) rate of 1.0, 0.5, 0.25, 0.1, 0.0 in the remaining five 10min intervals. The five sensors report 10min average concentrations in the same six 10min intervals as the source is emitting at a constant rate (this is just for convenience and is not required). The atmospheric dispersion model INPUFF (Petersen & Lavdas, 1986) was used to simulate the dispersion of the release, which includes computing average concentrations at the five sensors sites in the six 10min time intervals. These values were taken as the true concentrations at the five sites in the six time periods; that is, in terms of the notation introduced in Section 2.3,

$C(\mathbf{m}_j, t) = \hat{C}(\mathbf{m}_j, t)$ = the INPUFF predicted contaminant average concentration in the t -th time period, $t = 1, \dots, 6$, at sensor location \mathbf{m}_j , $j = 1, \dots, 5$.

Sensor data $\{c_{j,t} : j = 1, \dots, 5, t = 1, \dots, 6\}$ was then generated according to the truncated Gaussian data-model in (9) with mean $\hat{C}(\mathbf{m}_j, t)$ and variance $V(\hat{C}(\mathbf{m}_j, t))$ given by

$$V(\hat{C}(\mathbf{m}_j, t)) = (1\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t))^2. \quad (35)$$

Hence, the standard deviation is given by $\sqrt{1\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t)}$, indicating that measured average 10min concentration of around $1\text{E-}9$ and below are not distinguishable from zero, while higher concentration measurements have an approximated coeffi-

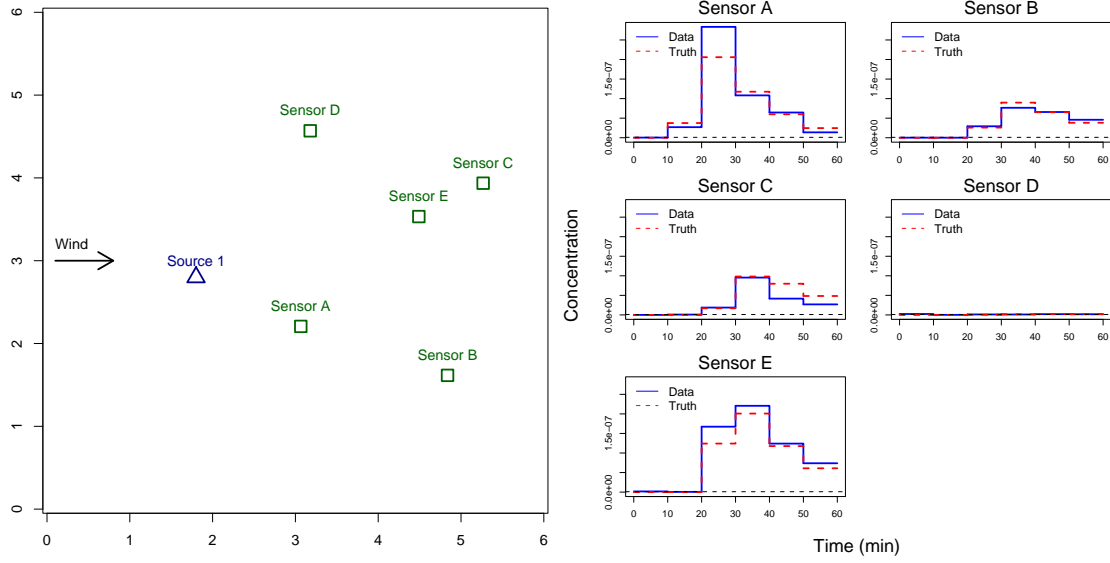


Figure 3: Left, the location of the stationary release source along with the five sensors. Right, the synthetic true 10min average concentration at the five sensor sites along with the synthetic observed concentrations.

cient of variation (CV) equal to 0.2 (20%). Figure 3 (right) shows the synthetic truth $\{\hat{C}(\mathbf{m}_j, t)\}$ at the five sensors along with the synthetic data $\{c_{j,t}\}$.

Finally, we note that the INPUFF model satisfies the additive factorization of the predictive concentration as given by (8). This leads to simplifications (and time-savings) in computations.

Initial MCMC at $t = 2$

From Figure 3 (right) we see that the first non-zero concentration is observed in the second 10min time interval at sensor A, a concentration of $2.7\text{E-}8$, with the remaining four sensors reporting zero concentrations (or rather, concentrations below detection level).

We now seek to start an initial MCMC sampler to sample from the posterior distribution of the unknown source location, \mathbf{x} , and the release rate in the first two 10min time interval, $\mathbf{s}_{1:2}$; that is, we seek to sample from $\pi_2(\boldsymbol{\theta}_{1:2})$, $\boldsymbol{\theta}_{1:2} = (\mathbf{x}, \mathbf{s}_{1:2})$. We assume a flat prior on the location of the source, as outlined in Section 2.3, and a prior on the release rate that assumes an unknown start (i.e., either in the first or the second time period) and then truncated Gaussian distribution for a non-zero release; see (10)–(12). In terms of the notation in Section 2.3, we take the initial non-zero release prior to be given by

$$f_1(s_{t*}) = f(s_{t*}) \text{ is } \text{Gau}(0, 20^2) \Big|_0^\infty,$$

which is also the prior we use for subsequent releases; that is, $f_2(s_2 | s_1) = f(s_1)$.

Hence, we assume that knowing s_1 has no value in determining s_2 *a priori* (a rather vague assumption). To summarize, the prior on $\boldsymbol{\theta}_{1:2}$ is given by

$$p(\boldsymbol{\theta}_{1:2}) = \begin{cases} f(s_1)f(s_2) & \text{if } t^* = 1, \\ f(s_2) & \text{if } t^* = 2. \end{cases}$$

We take the data-model, the likelihood, to be given by the product of the individual distributions in (9), yielding

$$p(\mathbf{c}_{1:2} | \boldsymbol{\theta}_{1:2}) = \prod_{t=1}^2 \prod_{j=1}^5 \varphi(c_{j,t}; \hat{C}(\mathbf{m}_j, t), (2\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t))^2) \Big|_0^\infty, \quad (36)$$

where $\varphi(c; \mu, V) \Big|_0^\infty$ is the density of a Gaussian distribution with mean μ and variance V , but restricted to the interval $(0, \infty]$. Note we have inflated the variance slightly by adding 1E-9 to the standard deviation used to generate the synthetic data; see (35). This mirrors reality, where the likelihood used in the MCMC sampler is just an approximation to the true (unknown) likelihood function.

The proposal distribution is a mixture of random-walk proposals and consist of either: (1) making a release rate change proposal, or (2) making a source location change proposal, or (3) making a joint release and location change proposal.

For the source location we use a random-walk on a lattice with a 0.1 horizontal/vertical distance between grid-locations:

Location Proposal

Let \mathbf{x} be the current location of the Markov chain, then:

- (1) Create the grid-point neighborhood set

$$\mathcal{N}_d(\mathbf{x}) \equiv \{\tilde{\mathbf{x}} : |x_1 - \tilde{x}_1| \leq d, |x_2 - \tilde{x}_2| \leq d, \text{ and } \tilde{\mathbf{x}} \neq \mathbf{x}\},$$

where $d > 0$ is a given neighborhood-size parameter, and recall that $\mathbf{x} = (x_1, x_2)$ and $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)$.

- (2) Generate the source location proposal $\tilde{\mathbf{x}} \sim q_2(\tilde{\mathbf{x}} | \mathcal{N}_d(\mathbf{x}))$, where

$$q_2(\tilde{\mathbf{x}} | \mathcal{N}_d(\mathbf{x})) = \frac{1}{|\mathcal{N}_d(\mathbf{x})|} I(\tilde{\mathbf{x}} \in \mathcal{N}_d(\mathbf{x})), \quad (37)$$

$|\mathcal{N}_d(\mathbf{x})|$ = the number of grid-points in $\mathcal{N}_d(\mathbf{x})$, and $I(\tilde{\mathbf{x}} \in \mathcal{N}_d(\mathbf{x})) = 1$ if $\tilde{\mathbf{x}} \in \mathcal{N}_d(\mathbf{x})$, otherwise equal to 0.

The size of the neighborhood $\mathcal{N}_d(\mathbf{x})$, given by d , affects the efficiency of the location proposal. If d is too small, the resulting chain does not mix well and in addition can also get “stuck” sampling in the vicinity of a local posterior mode. If d is too large, large number of proposals gets rejected, but the chain is less likely to get stuck around a local posterior mode. The approach we take is to select randomly the neighborhood size d among three values, $d = 0.1, 0.3, 2$, with probability of selecting each equal to $2/7, 4/7, 1/7$, respectively. Hence, if a source location proposal is made, a neighborhood-size parameters d is first drawn randomly, then a location is selected randomly from $\mathcal{N}_d(\mathbf{x})$.

The benefits of working with a source location lattice is in terms of reduced number of INPUFF runs needed, as one can store the results for each grid-location by storing the values $\{\hat{G}_{\mathbf{x},t}(\mathbf{m}_j, t')\}$ for each grid-location \mathbf{x} .⁴ The drawback of the lattice approach is that we cannot distinguish between source locations within a 0.1×0.1 pixel. In practice, the resolution of the lattice can be linked to the accuracy of the dispersion simulation program; a less accurate dispersion simulator can work on a coarser grid.

The proposal distribution for the source release rates, $\mathbf{s}_{1:2}$, is slightly more involved and is a two-step mixture; either propose a change in the start time of the release or propose a change to the current non-zero release rates (or propose both at the same time).

A change in the start time (t^*) is simply accomplished via random-walk to a nearest neighbor. Since $\mathbf{s}_{1:2}$ is only of length two, it is just an issue if the release started in the first time interval or the second time interval. Let $\mathbf{s}_{1:2} = (s_1, s_2)$ be the current release rate. The change of start-time proposal is given by:

Release-Rate Start-Time Proposal

- (1) If $t^* = 1$, that is if $s_1 > 0$, then $\tilde{t}^* = 2$ is proposed with

$$\tilde{\mathbf{s}}_{1:2} = (\tilde{s}_1 = 0, \tilde{s}_2 = s_2),$$

yielding $q_2(\tilde{t}^* = 2, \tilde{\mathbf{s}}_{1:2} | t^* = 1, \mathbf{s}_{1:2}) = 1$.

- (2) If $t^* = 2$, that is if $s_1 = 0$, then $\tilde{t}^* = 1$ is proposed with

$$\tilde{\mathbf{s}}_{1:2} = (\tilde{s}_1, \tilde{s}_2 = s_2), \quad \text{where } \tilde{s}_1 \sim \text{Gau}(0, 5^2) \Big|_0^\infty,$$

yielding $q_2(\tilde{t}^* = 1, \tilde{\mathbf{s}}_{1:2} | t^* = 2, \mathbf{s}_{1:2}) = \varphi(\tilde{s}_2; 0, 5^2) \Big|_0^\infty$.

A change to the non-zero release rates is proposed via random-walk as follows:

⁴Actually, we are able to get $\{\hat{G}_{\mathbf{x},t}(\mathbf{m}_j, t') : \mathbf{x} = \text{all grid points}, t \leq t'\}$ in a single ‘reverse’ INPUFF-run for each value of (\mathbf{m}_j, t') ; $j = 1, \dots, 5, t' = 1, 2$. Hence, this requires only a total of 10 INPUFF runs.

Non-Zero Release-Rate Proposal

- (1) Propose to change n_c non-zero release rates, where

$$n_c = 1 + \Delta_c, \quad \Delta_c \sim \text{Bin}(N_c, \pi_c),$$

where $\text{Bin}(N_c, \pi_c)$ denotes a binomial distribution on $\{0, \dots, N_c\}$, $N_c = t - t^*$ and $\pi_c \in (0, 1]$ is the rate parameter. Note, if $t = t^* = 2$, then $n_c = 1$, but if $t^* = 1$, either one or two release rates are changed according to the rate parameter π_c .

- (2) Given the number of release rates to change (n_c), select randomly among the non-zero release rates which one to change and let $\{t_{c,j} : j = 1, \dots, n_c\}$ index the selected time periods.
- (3) For $j = 1, \dots, n_c$, make the random-walk proposal,

$$\tilde{s}_{t_{c,j}} \sim \text{Gau}(s_{t_{c,j}}, \tau_j^2) \Big|_0^\infty,$$

where the standard deviation τ_j specifies the “step-size”. The τ_j ’s are selected randomly from the set $\{1, 3, 9\}$ with probability $\{2/7, 4/7, 1/7\}$, respectively. Hence, each random-walk is carried out with different step-size.

The proposal density is then given by

$$q_2(\tilde{\mathbf{s}}_{1:2} \mid \mathbf{s}_{1:2}) = \prod_{j=1}^{n_c} \varphi(\tilde{s}_{t_{c,j}}; s_{t_{c,j}}, \tau_j^2) \Big|_0^\infty,$$

and note that we consider n_c , $\{t_{c,j}\}$, and $\{\tau_j\}$ fixed; that is, the reverse proposal density $q_2(\mathbf{s}_{1:2} \mid \tilde{\mathbf{s}}_{1:2})$ is computed with the same numbers.

When a decision is made to make a source release change, a random draw is made as to: (1) make a change to the start-time, (2) make a change to the non-zero release rates, or (3) make a simultaneous change to the start-time and non-zero releases. The probability assigned to these three types of proposals is 1/12, 10/12 and 1/12. That is, most of the time a non-zero release rate proposal is made.

The MCMC proposal step then alternates in a random fashion between making (1) a source location proposal, (2) making a source release rate proposal, or (3) make both source location and release rate proposals. An equal probability was assigned to the three different types.

Six different MCMC samples, each of size 10,000, were generated using the above proposal process. All six chains were initialized with the release rate $\mathbf{s}_{1:2}^{(0)} = (0.1, 0.1)$, but at six different locations:

$$\mathbf{x} \in \{(4, 1), (4, 3), (4, 5), (1, 1), (1, 2), (1, 5)\}.$$

The acceptance rate for each chain was about 20% (this low acceptance rate is expected as the proposal process has a number of save-guard sub-proposals steps that have a very low change of being accepted when performed, but can potentially move the chain across low-probability barriers).

For the first 500 iterations ($i = 1, \dots, 500$), the likelihood was taken to be given by

$$p(\mathbf{c}_{1:2} | \boldsymbol{\theta}_{1:2})^{1/T_i},$$

where $T_i = 1 + (10 - i \times (10/500))$ and often referred to as the annealing temperature; see, for example Liu (2001), chapter 10. This causes the true likelihood (and hence, the data) to be brought in “slowly” as a high value of T results in a “flatter” likelihood (heated likelihood). This annealing process is well known technique to escape from a bad initial values, for example, one that is located in the vicinity of a local posterior mode of a low probability mass.

Figure 4 summarizes the MCMC output from the chain initialized at location (4,3) and the chain initialized at location (1,2) — the first 500 iterations were discarded (recall those use the “heated” likelihood). Both chains quickly fixates on realizations with $s_1 = 0$ (which was how the synthetic data was generated), but the data seems to provide little information on the release rate in the second time period, as it is seen to vary widely. Most realizations for the source location form a half-circle upwind from the sensor reporting the only non-zero concentration. The second chain, initialized at source location (1,2), generates in the beginning source location realizations that are clustered together in the lower-left corner. This cluster of realizations has lower posterior probability compared to the main cluster, as can be seen from the trace plot of the log-posterior (a log-posterior difference of about 5 translates into posterior density ratio of about 150).

The six chains were combined to form a single posterior sample, with the first 1/3 of each chain discarded as a burn-in period. Figure 5 shows two maps of the marginal posterior distribution of the source location. It shows a half-circle shaped distribution upwind from the only sensor reporting a non-zero concentration. The true location of the source is at the edge of the posterior distribution.

Figure 6 shows the posterior marginal distribution of the release rate in the two time periods. The release rate for the first time period is estimated to have $p(s_1 = 0 | \mathbf{c}_{1:2}) = 0.89$; that is, most likely no release in the first time period (which is the case). However, the data is not very informative for the release in the second time interval, and the marginal posterior distribution for s_2 is very close to the prior distribution; Figure 6 (left). It is often more informative to look at the posterior release rate conditional on a given location. Figure 6 shows the expected (average) non-zero release rate in the second time period for different potential source locations (those with non-zero posterior probability). As can be seen, locations further away from the sensors are associated with higher release rates than those that are closer to the sensors.

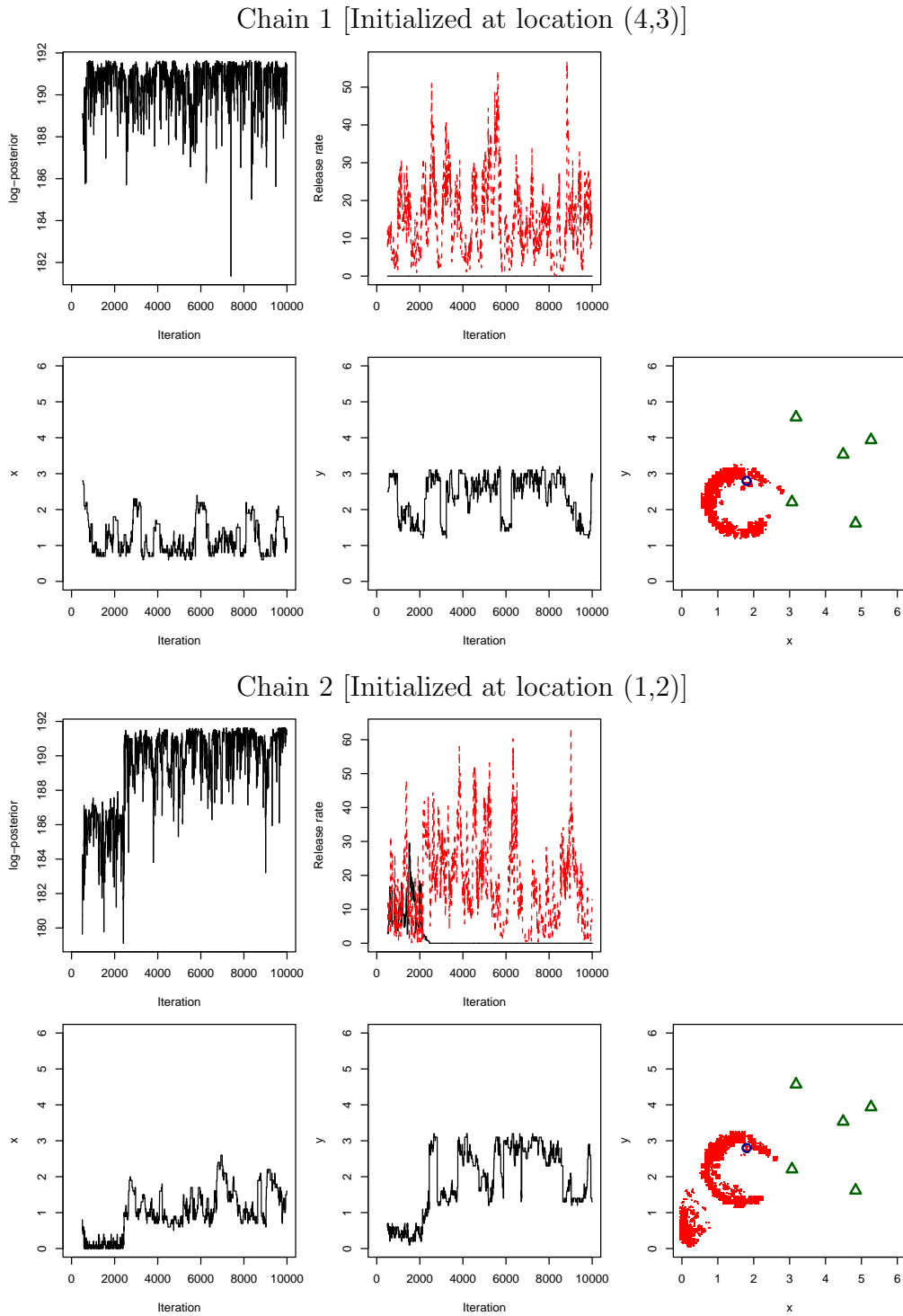


Figure 4: MCMC summary for two chains. Shown is the trace of the log-posterior distribution (up to an unknown additive constant), the trace of the release rate parameters, the trace of the x and y components of the source location, and finally a plot of the sampled source locations along with the location of the sensors and the true location of the source.

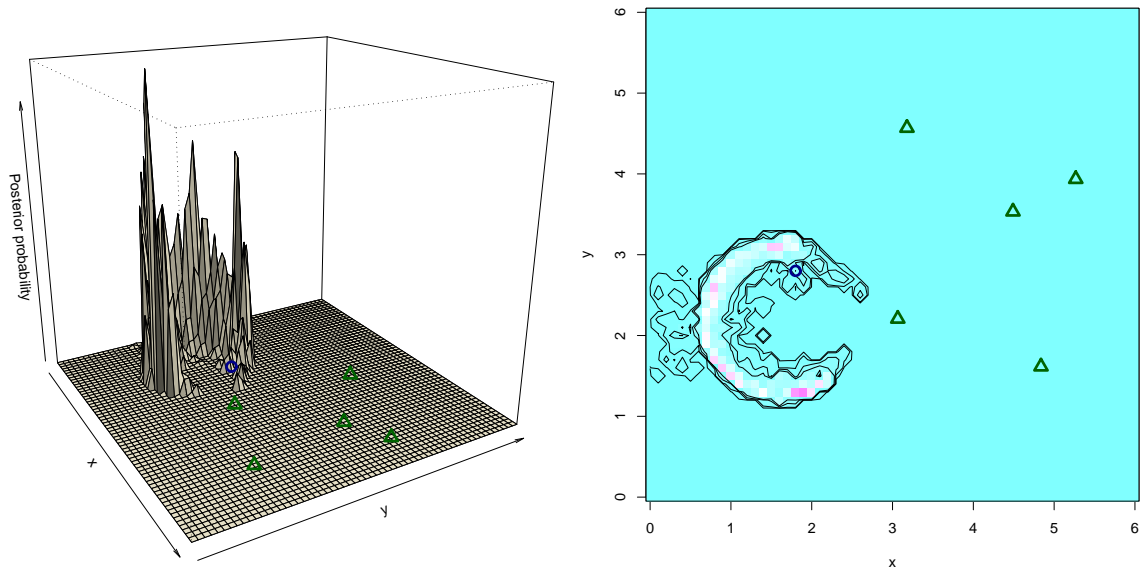


Figure 5: Left, a 3D perspective plot of the marginal posterior distribution of the source location. Right, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets).

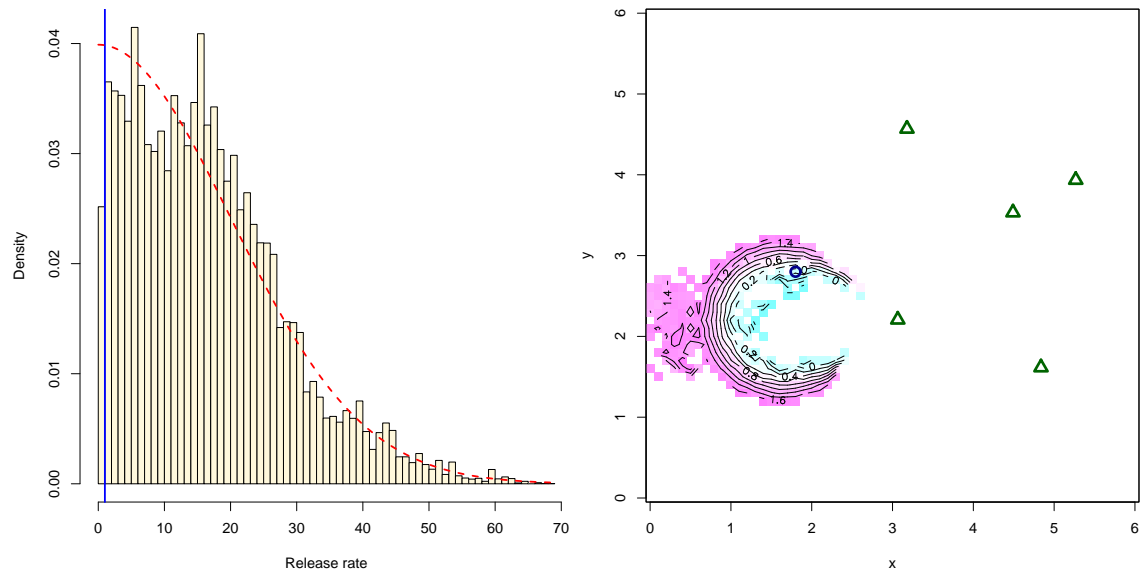


Figure 6: Left, a histogram of the posterior samples for the release rate in the second time period. The true release rate is indicated with a (blue) vertical line and the prior distribution is shown as a (red) dotted line. Right, the expected (average) non-zero release rate, \log_{10} -transformed, in the second time period conditional on location (i.e., \log_{10} average release-rate at each location pixel).

MCMC at $t = 3$

As new data arrives in the third time period ($t = 3$), one can carry out a new MCMC for posterior inference or use SMC, using the MCMC sample from $t = 2$ as the initial sample. We shall now carry out a MCMC posterior sampling for $t = 3$ (starting from scratch), but later one we shall use SMC for the same purpose.

We applied the same proposal process at $t = 3$ as at $t = 2$, with obvious extensions to make it applicable for three time periods. A short initial run was carried out to fine-tune the proposal distributions (i.e., step-size of random-walk samplers, etc.), then six different chains were sampled, as in the case for $t = 2$.

Figure 7 summaries the result for two of the six chains in the same way as in Figure 4. There is considerable more non-zero concentration sensor-observations available at $t = 3$ that yield stronger posterior information. We see that one of the chains in Figure 7 quickly converges while the other one needs approximately 4,000 iterations to stabilize.

We combined the samples from the six chains after discarding the first half of each chain (a rather conservative approach). Figure 8 shows the marginal posterior distribution of the source location and the marginal distribution of the source release-rate in the second time period (s_2). There is a much stronger posterior knowledge about the source location at this time. Similarly, the marginal posterior distribution for the release rate at $t = 2$ is rather peaked with the true release rate close to the posterior peak. The release rate at $t = 1$ is estimated to be equal to 0 with 99.97% probability. However, not much is known about the release rate at $t = 3$ (as expected).

SMC

We shall now carry out SMC for $t = 3, \dots, 6$ using the last 6,667 MCMC realizations (the first 3,333 discarded as a burn-in period) from each of the six chains at $t = 2$ as the initial posterior sample;

$$\Theta_{1:2} = \{\boldsymbol{\theta}_{1:2}^{(i)} : i = 1, \dots, 40,002\},$$

where the realizations are all of equal weight. We shall use Pitt's and Shephard's (P&S) modification of Gordon's bootstrap filter, as outlined in Section 4.3, with the addition of performing MCMC perturbation within each SMC cycle, as outlined in Section 4.5 on hybrid methods. The SMC-MCMC algorithm applied is given in Table 5, with details on the various proposal distribution used to follow.

The likelihood, $p(\mathbf{c}_t | \boldsymbol{\theta}_{1:t})$, is as in (9) and (36);

$$p(\mathbf{c}_t | \boldsymbol{\theta}_{1:t}) = \prod_{j=1}^5 \varphi(c_{j,t}; \hat{C}(\mathbf{m}_j, t), (2\text{E-}9 + 0.2 \times \hat{C}(\mathbf{m}_j, t))^2) \Big|_0^\infty.$$

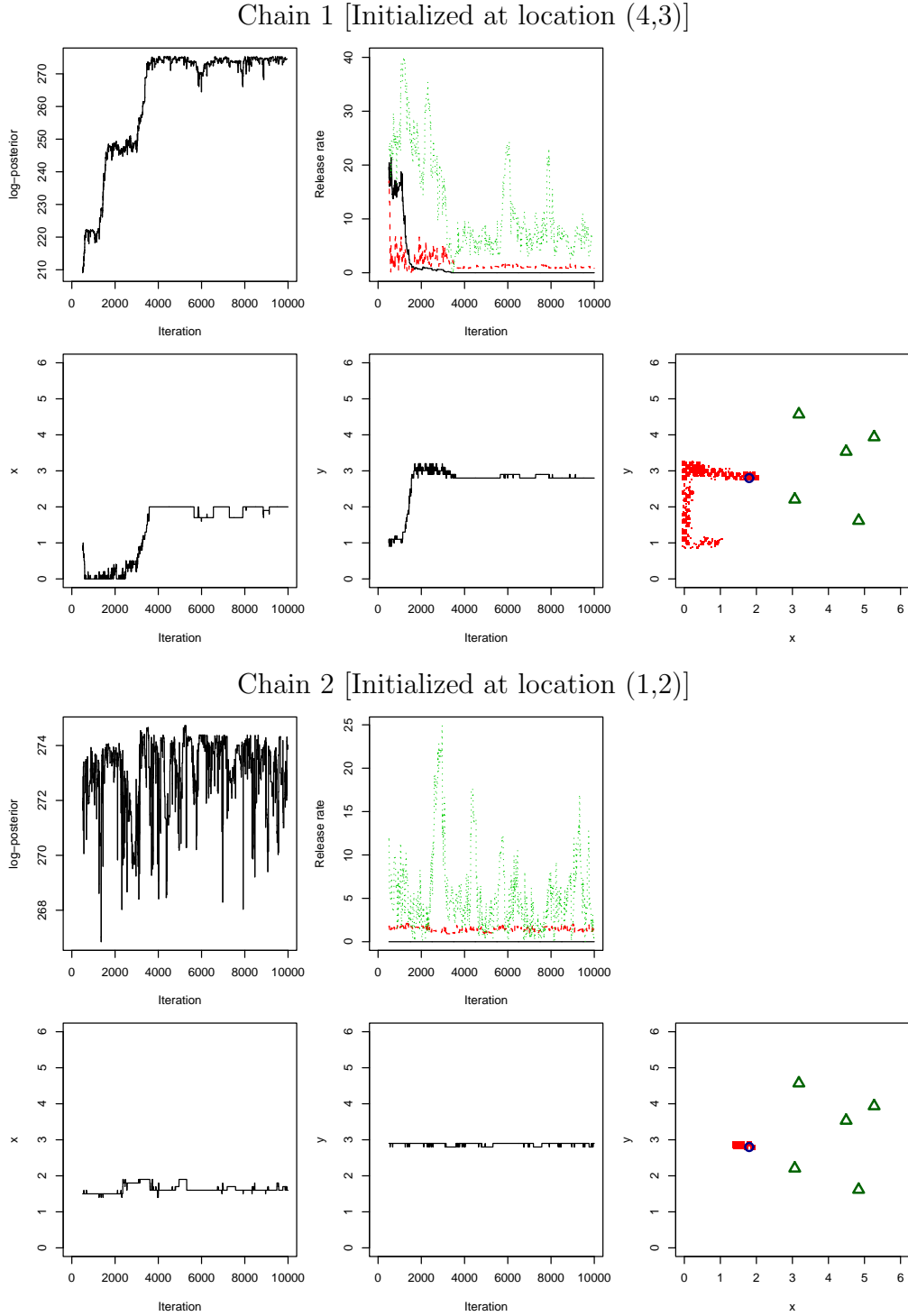


Figure 7: MCMC summary for two chains of six at $t = 3$; see Figure 4 for the content of each plot.

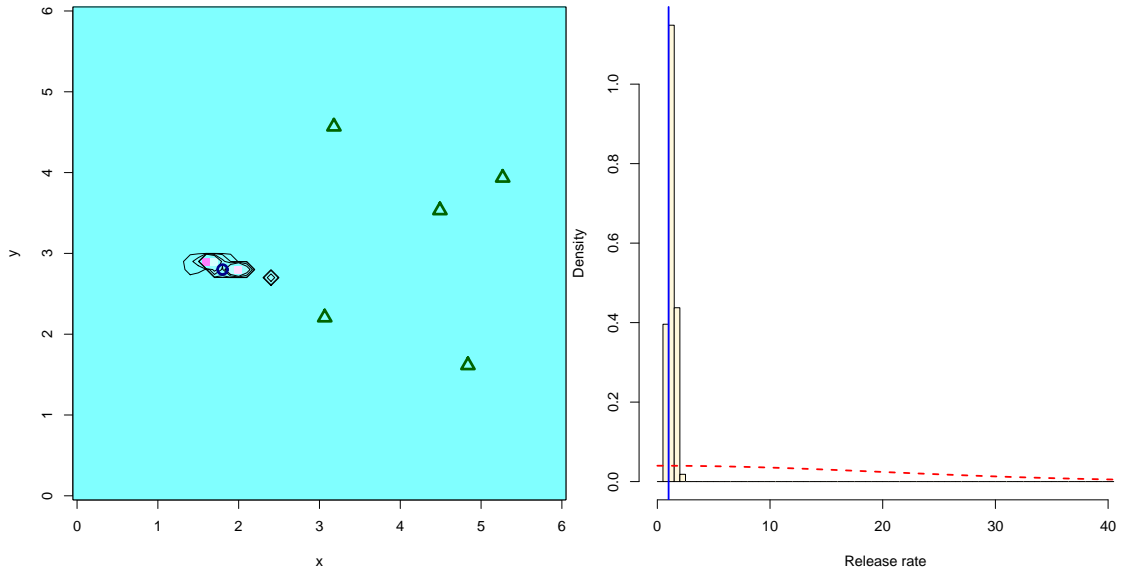


Figure 8: Left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). Right, a histogram of the posterior samples for the release rate in the second time period, with the true release rate shown as a (blue) vertical line and the prior distribution shown as a (red) dotted line.

Similarly, the conditional prior for $\boldsymbol{\theta}_t$, $p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1})$, is as in the MCMC case for non-zero releases, and given by

$$p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}) = p(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}) = \begin{cases} \varphi(s_t; 0, 20^2) & \text{if } \mathbf{x}_t = \mathbf{x}_{t-1}, \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

Note, this is a very vague conditional prior on the release rate as it does not depend on $\mathbf{s}_{1:t-1}$ at all.

The proposal distribution $q_t(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1})$ in Table 5 is given by

$$q_t(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{1:t-1}) = q_t(\mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}) = \begin{cases} \varphi(s_t; s_{t-1}, 10^2)|_0^\infty & \text{if } \mathbf{x}_t = \mathbf{x}_{t-1}, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, it proposes no change in location (as expected and in accordance to our model) and then s_t is generated from a Gaussian distribution with mean s_{t-1} and standard deviation 10, and constrained to the interval $[0, \infty)$.

For the MCMC perturbation step in Table 5 we used a similar proposal process as in the MCMC-only application previously for $t = 2$ and $t = 3$. That is, a random choice is made to carry on: (1) a proposal to change the source release, (2) a proposal to change the source location, or (3) both. The probability assigned to these three proposals is 1/6, 4/6, and 1/4, respectively.

The source-release proposal consists of a random-walk proposal for a selected source-release time period. Let $\boldsymbol{\theta}_{1:t}^{(i,j)} = (\mathbf{x}^{(i,j)}, \mathbf{s}_{1:t}^{(i,j)})$ be the current value of the Markov chain, then:

Release-Rate Proposal

- (1) Select a time period \check{t} from $\{t-2, t-1, t\}$, with probability 1/4, 2/4, and 1/4 of selecting each period, respectively.
 - (2) Generate $\check{s}_{\check{t}} \sim \text{Gau}(s_{\check{t}}^{(i,j)}, \sigma_{\check{t}}^2)|_0^\infty$ and put the remaining release rates of $\check{\mathbf{s}}_{1:\check{t}}$ identical to those of $\mathbf{s}_{1:\check{t}}^{(i,j)}$.
-

The standard deviations (SD) used in the release-rate proposal above were given by,

$$\sigma_k = 0.75 \times \{\text{the empirical SD of } \{\check{s}_k^{(i,0)} : i = 1, \dots, N\}\},$$

but never taken less than 0.1².

The proposal for the source location was taken to be a random-walk to a grid-point within a horizontal or vertical distance of 0.2 from the current location; that is, as the proposal given in (37).

The SMC results are summarized in Figures 9–11, for the time periods 1–3, 1–4, and 1–6, respectively. Each figure shows the marginal posterior distribution of the source location and the marginal posterior distribution the release rate for the three most recent time periods in each case. As expected, as more data is gathered, the marginal posterior distribution of the source locations narrows around the true location of the source. Similarly, as more data is processed, we gain better knowledge about the source release-rate history. Note how the posterior distribution of the release rate in the most recent time period in each case gets more informative (narrower) at later time periods. This is due to a narrower posterior distribution for the source location, which limits what the potential release rates in the latest periods could be, given the data.

MCMC versus SMC

Both MCMC and SMC samples were generated for posterior inference at $t = 3$; see Figure 8 and Figure 9, respectively. We notice a slight difference in the shape of the highest posterior density (HPD) regions constructed for the source location based on the two methods. However, the extent of the HPDs regions are very similar for both methods. The SMC-based posterior distribution of s_2 in Figure 9 seems to be slightly narrower than the one shown in Figure 8 and based on the MCMC sample. In general, we believe that the SMC sample gives a better representation of the posterior than the MCMC sample; the SMC sample consist of a slightly larger number of realizations (about 40,000 versus 30,000), but more importantly, it is a better mixed sample since *each* SMC realizations is independently perturbed via 10 MCMC iterations.

One might get the impression that the SMC algorithm needs considerable more computation time than the MCMC algorithm since, in addition to extending the past SMC realizations from time $t = 2$ to $t = 3$, it performs 10 MCMC iteration for each SMC realization (a total of 400,000 MCMC iterations). However, this is not the case. As the SMC is not a sequential algorithm (like the MCMC algorithm), one can take advantage of highly optimized vectorized computer operations that operate on all the realizations at once. In fact, our current (serial) prototype implementation of the SMC algorithm ran faster than the MCMC implementation. In addition, it is relatively easy to implement the SMC algorithm to effectively use a computer with a large number of CPUs (e.g., a Linux cluster), while this is not the case for the MCMC algorithm.

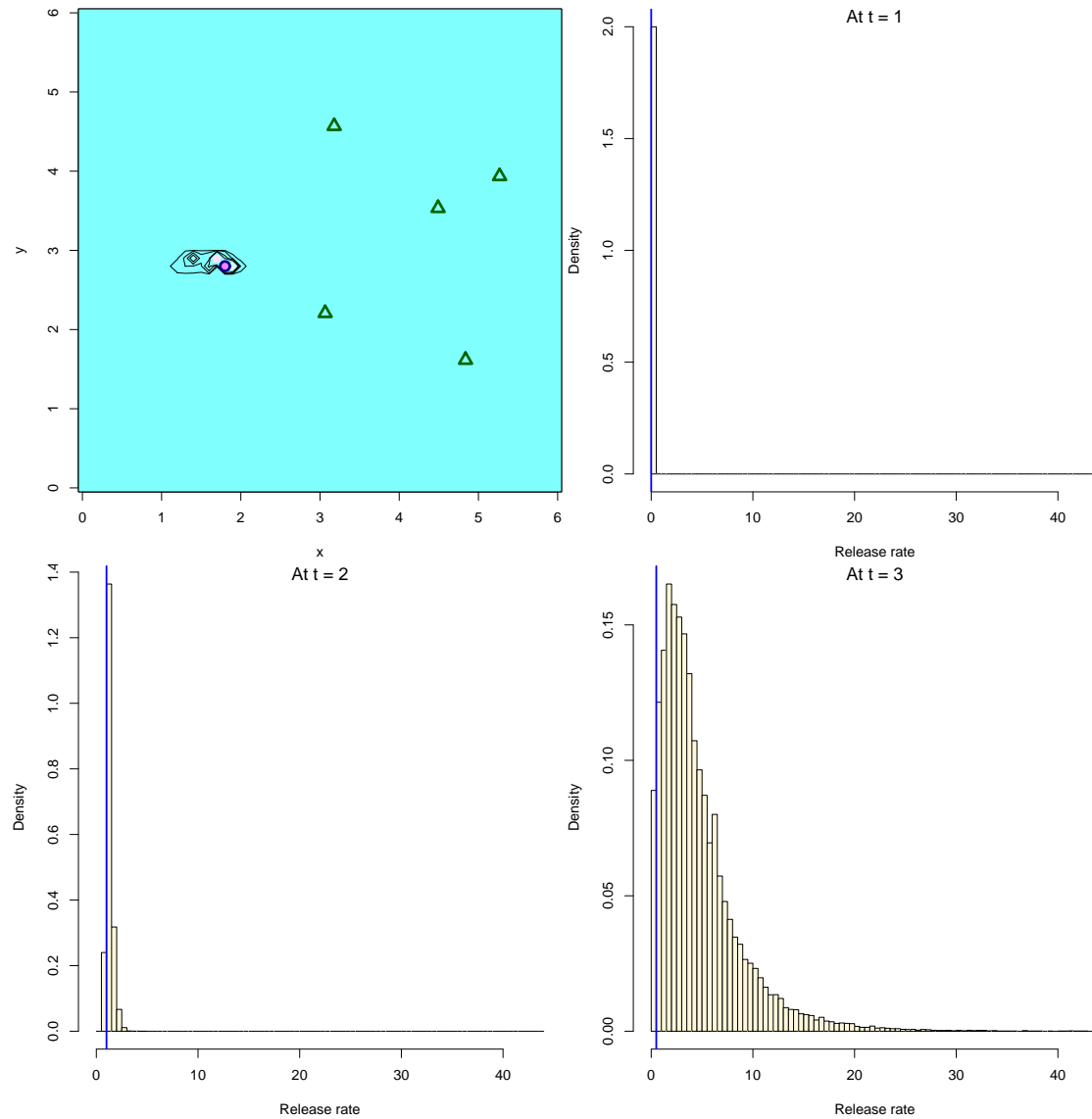


Figure 9: SMC posterior inference after processing data from time periods 1–3. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 1, 2, 3$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

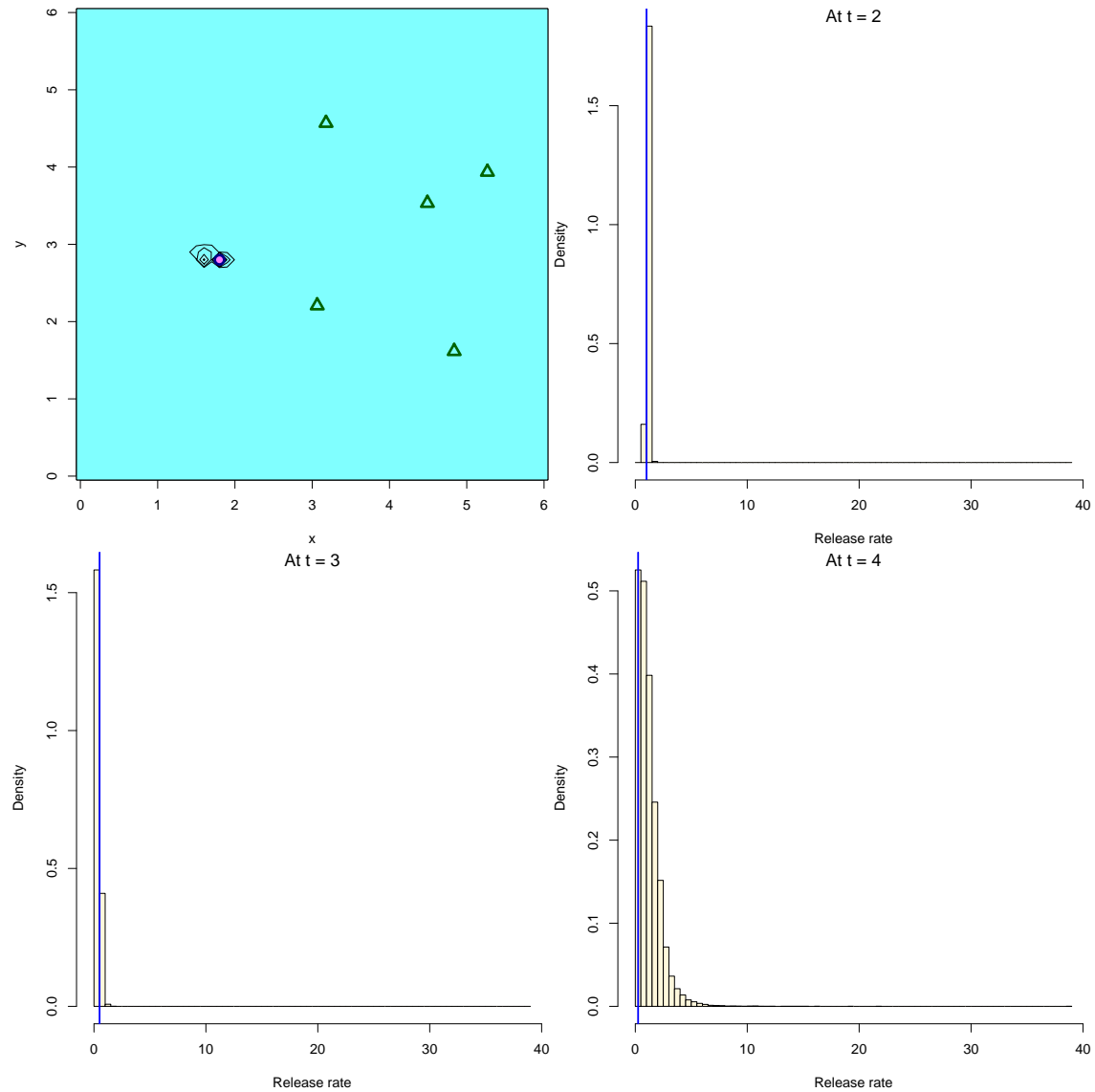


Figure 10: SMC posterior inference after processing data from time periods 1–4. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 2, 3, 4$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

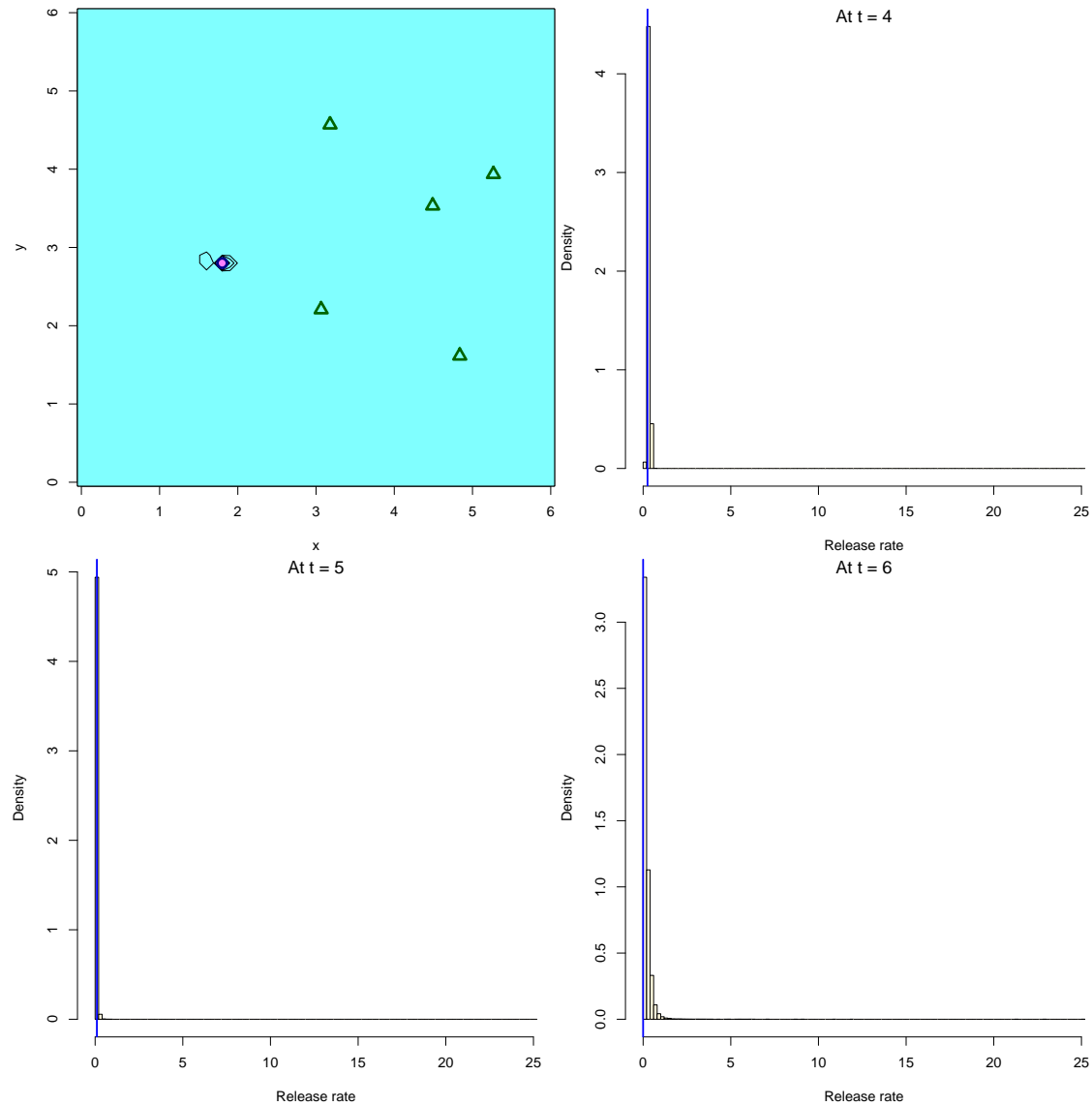


Figure 11: SMC posterior inference after processing data from time periods 1–6. Top-left, a 2D level plot of the marginal posterior distribution of the source location (color scheme: cyan = low, magenta = high) along with contours showing the regions containing the 90%, 95%, 99%, and 99.9% of the highest posterior probability density (HPD credible sets). The remaining plots show a histogram of the posterior samples for the release rate at $t = 4, 5, 6$, with the true release rate shown as a (blue) vertical line (note different horizontal scale).

Table 5: SMC-MCMC Algorithm for Atmospheric Event Reconstruction.

The hybrid SMC-MCMC algorithm used to generate samples from the posterior at times $t = 3, \dots, 6$ in the atmospheric reconstruction application. Details on proposal distributions provided in text.

Initial Sample: Start with the initial, equal-weighted sample $\{\theta_{1:2}^{(i)} : i = 1, \dots, N\}$, $N = 40,002$, derived from the initial MCMC samples.

For $t = 3, \dots, 6$: (Looping through the time periods)

Proposal Weights (P&S): For $i = 1, \dots, N$:

- (1) Put $\hat{\theta}^{(i)} = 0$.
- (2) Compute $v_{1:t-1}^{(i)} = p(\mathbf{c}_t | \theta_{1:t-1}^{(i)}, \hat{\theta}_t^{(i)})^2$. [“heated” likelihood.]

Extending to time t : For $i = 1, \dots, N$:

- (1) Sample $\tilde{I}_i \in \{1, \dots, N\}$ with $p(\tilde{I}_i = j) \propto v_{1:t-1}^{(j)}$; $j = 1, \dots, N$.
- (2) Generate $\tilde{\theta}_t^{(i)} \sim q_t(\theta_t | \theta_{1:t-1}^{(\tilde{I}_i)})$ and let $\tilde{\theta}_{1:t}^{(i)} \equiv (\theta_{1:t-1}^{(\tilde{I}_i)}, \tilde{\theta}_t^{(i)})$.
- (3) Compute the importance-sample weight

$$\tilde{w}_{1:t}^{(i)} = \frac{p(\mathbf{c}_t | \tilde{\theta}_{1:t}^{(i)})p(\tilde{\theta}_t^{(i)} | \tilde{\theta}_{1:t-1}^{(i)})}{q_t(\tilde{\theta}_t^{(i)} | \theta_{1:t-1}^{(i)})} \frac{1}{v_{1:t-1}^{(i)}}. \quad [\text{recall, } w_{1:t-1}^{(i)} \propto 1.]$$

MCMC Perturbation: For $i = 1, \dots, N$:

Selection: Select $\tilde{I}_i \in \{1, \dots, N\}$ with $p(\tilde{I}_i = j) \propto \tilde{w}_{1:t}^{(j)}$; $j = 1, \dots, N$, and put $\theta_{1:t}^{(i,0)} = \tilde{\theta}_{1:t}^{(\tilde{I}_i)}$.

MCMC Loop: For $j = 1, \dots, B$: [$B = 10$ used.]

- (1) Propose $\check{\theta}_{1:t} \sim q_t(\check{\theta}_{1:t} | \theta_{1:t}^{(i,j-1)})$.
- (2) Compute the M-H ratio

$$\rho_t(\check{\theta}_{1:t}; \theta_{1:t}^{(i,j-1)}) = \frac{p(\mathbf{c}_{1:t} | \check{\theta}_{1:t})p(\check{\theta}_{1:t})q_t(\theta_{1:t}^{(i,j-1)} | \check{\theta}_{1:t})}{p(\mathbf{c}_{1:t} | \theta_{1:t}^{(i,j-1)})p(\theta_{1:t}^{(i,j-1)})q_t(\check{\theta}_{1:t} | \theta_{1:t}^{(i,j-1)})}.$$

- (3) Generate $u \sim \text{Unif}[0, 1]$ and put $\theta_{1:t}^{(i,j)} = \check{\theta}_{1:t}$ if $\rho_t(\check{\theta}_{1:t}; \theta_{1:t}^{(i,j-1)}) > u$, otherwise put $\theta_{1:t}^{(i,j)} = \theta_{1:t}^{(i,j-1)}$.

Collect: Put $\theta_{1:t}^{(i)} = \theta_{1:t}^{(i,B)}$, then $\{\theta_{1:t}^{(i)} : i = 1, \dots, N\}$ is an equal-weighted sample from $\pi_t(\theta_{1:t})$.

References

- Andrieu, C., De Freitas, N., Doucent, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50, 5–43.
- Arulampalam, M., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50, 174–188.
- Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*. Wiley.
- Doucet, A., de Freitas, J. F. G., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag.
- Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian Data Analysis* (Second Edition ed.). Boca Raton, Florida: Hapman and Hall/CRC.
- Gilks, W. R. & Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society B*, 63, 127–146.
- Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. E. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.
- Godsill, S. & Clapp, T. (2001). Improvement strategies for Monte Carlo particle filters. In A. Doucent, N. de Freitas, & N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice* (pp. 139–158). New York: Springer.
- Gordon, N. J., Salmon, D. J., & Smith, A. F. M. (1993). A novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, 140, 107–113.
- Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: Springer.
- MacEachern, S. N., Clyde, M., & Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, 27, 251–267.
- Petersen, W. & Lavdas, L. (1986). Inpuff 2.0: A multiple source gaussian puff dispersion algorithm — user’s guide. Technical Report EPA/600/8-86/024, EPA.
- Pitt, M. K. & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 23, 356–359.
- Pitt, M. K. & Shephard, N. (2001). *Sequential Monte Carlo methods in practice*, chapter Auxiliary variable based particle filters. New-York: Springer-Verlag.

West, M. & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models (Second Edition)*. New York: Springer-Verlag.